

UNIVERSIDADE FEDERAL DO PARANÁ

MARCELINO ULICA ABEL

CLASSIFICAÇÃO DE SÉRIES TEMPORAIS POR MEIO DA
TRANSFORMADA *SHAPELET*

CURITIBA PR

2018

MARCELINO ULICA ABEL

CLASSIFICAÇÃO DE SÉRIES TEMPORAIS POR MEIO DA
TRANSFORMADA *SHAPELET*

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática, no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Fabiano Silva.

Coorientador: Prof. Dr. Willian Zalewski.

CURITIBA PR

2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

A139c

Abel, Marcelino Ulica

Classificação de Séries Temporais por meio da Transformada Shapelet /
Marcelino Ulica Abel. – Curitiba, 2018.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas,
Programa de Pós-Graduação em Informática, 2018.

Orientador: Fabiano Silva – Coorientador: Willian Zalewski. -

1. Análise de séries temporais - programas de computador 2. Algoritmos
computacionais 3. Mineração de dados (Computação) . I. Universidade
Federal do Paraná. II. Silva, Fabiano. III. Título.

CDD: 519.55

Bibliotecária: Vanusa Maciel - CRB - 9/1928



MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **MARCELINO ULICA ABEL** intitulada: **Classificação de Séries Temporais por meio da Transformada Shapelet**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 16 de Agosto de 2018.

FABIANO SILVA

Presidente da Banca Examinadora (UFPR)

MYRIAM REGATTIERI DE BIASE DA SILVA DELGADO

Avaliador Externo (UTFPR)

EDUARDO JAQUES SPINOSA

Avaliador Interno (UFPR)

WILLIAN ZALEWSKI

Coorientador - Avaliador Externo (UNILA)



*Em
memoria ao meu pai.
Antônio Abel*

Além disso, filho meu, sê avisado. De
fazer livros não há fim; e o muito estudar
é enfado da carne.

Ecl. 12:12.
"Recomendação da Dona Filomena,
minha mãe".

A coragem é a primeira das qualidades
humanas porque garante todas as outras.

Aristóteles

Agradecimentos

Ao meu Deus que é onipotente e onisciente de todas coisas, pelos cuidados e proteção em todos os momentos pelos quais passei.

Ao professor Fabiano Silva pela dedicação com a qual conduziu a orientação deste trabalho, pela compreensão em todas as etapas do desenvolvimento, por ter acreditado que era possível realizar este trabalho quando eu já não tinha mais forças para prosseguir, Ele contornou as minhas maiores debilidades, de modo que qualquer problema advir eu estaria preparado para solucionar, muito obrigado professor por compartilhar sua experiência durante este tempo de estadia em Curitiba-PR com este propósito.

Ao professor Willian Zalewski pela coorientação deste trabalho, e sendo o mentor da maior base fundamental desta pesquisa, pela aceitação e disponibilidade nos momentos mais cruciais e agir de modo preciso como um professor que sente a necessidade de demonstrar que é possível gerar resultados significantes com apenas pequenas mudanças em tarefas não menos triviais de serem realizadas.

A Simone F. Ramos, pelos ajustes, correções de ortografias (PT-BR) e o apoio demonstrado, serei eternamente grato.

A minha querida mãe Filomena Cambuende pela educação oferecida e sempre com aquela voz mansa e humilde me incentivando nas minhas escolhas, e cada telefonema recebido ela dizia "estas sempre nas minhas orações meu filho", e que se tudo isto não se concretizasse, haveriam sempre luzes que me guiarão de volta para casa. De fato Ela é a carta que ainda não escrevi.

Aos meus familiares, especialmente meu irmão Gabriel Chonguende Abel, que tem se dedicado muito em cuidar de nossa família desde que sai de casa com objetivo de estudar, e em muitos casos ele sempre se sacrificou pelo bem de qualquer um, fazendo o papel de um verdadeiro irmão, tio e amigo, com sinceridade ele sabia o que perguntar e como responder, tanto para um adulto quanto para uma criança.

Ao Josias Alfredo Kaveto, meu irmão que a vida me deu, por sempre estar presente e dividindo suas experiências e conquistas, planejando comigo vários projetos que num futuro tão próximo sairão do papel.

Ao nascimento da minha filha Heloise Abel que acabou sendo uma das grandes mudanças e prioridades da minha vida. Agradeço imenso pela Emília Vieira pela parceira e cuidar da nossa filha de um modo incrível, confrontando grandes desafios e sacrifícios em cuidar da Heloise, o meu muito obrigado.

Ao Laboratório de Inteligência Artificial e Métodos Formais (LIAMF) por possibilitar o desenvolvimento deste trabalho. Em especial aos colegas Mariane Cassenote, Bruno Zanette, Danielle Ivancheche, Clariane Menezes, e a todos meus amigos do DINF (departamento de informática) especialmente os colegas da Pós graduação (mestrandos e doutorandos) pela convivência e trocas de experiências e as discussões em cada tempo livre no nosso espaço tradicional para tomar café.

Resumo

O aumento dos dados coletados ao longo dos últimos anos despertou o interesse de vários pesquisadores em diferentes áreas de conhecimento concernente à análise de dados. Consequentemente, algoritmos de mineração de dados (MD) são utilizados para extração de padrões com o propósito de tomada de decisão. Dentre as tarefas de MD a classificação de séries temporais (CST) tem sido fortemente destacada. Neste contexto, a CST, mediante os algoritmos estado-da-arte e as técnicas para representação das séries temporais, consiste em tarefas não triviais, pois requer o conhecimento de domínio e de técnica para representá-las. Este trabalho apresenta uma avaliação experimental ampla da extensão da transformada *shapelet* (ETS) proposta por [Zalewski et al. \(2016\)](#), com base em outros modelos de classificação, e a proposta de uma nova ETS, visando a representação de séries temporais, extração de atributos relevantes e uma nova transformação dos dados sem perder informações importantes para inferências. Estas técnicas baseiam-se na eliminação de redundâncias entre as *shapelets*, o grau de correlação e autocorrelação entre elas, bem como a ordenação das *shapelet* de acordo com alguma medida de importância induzida nos algoritmos propostos. Após a separação das *shapelets* foram construídos modelos preditivos, baseados em *distâncias*, *probabilidades*, *procura e otimização*, com a finalidade de mostrar que a transformada *shapelet* (TS) proposta por [Lines et al. \(2012\)](#), em combinação com alguns algoritmos clássicos de seleção e agregação de atributos, resulta na redução e simplificação da complexidade da série temporal, permitindo que qualquer modelo preditivo adotado, sem a necessidade de ajustes de parâmetros, pode prover maior nível de desempenho em termos de acurácia. Conduzimos nossos experimentos em 25 conjuntos de dados extraídos do repositório UCR ([Chen et al., 2015](#)), para avaliação do desempenho dos modelos criados e dos métodos propostos. Após as análises dos resultados obtidos foram constatados desempenhos promissores em cada ETS utilizada.

Palavras-chave: classificação, séries temporais, *shapelet*, mineração de dados aprendizado de máquina.

Abstract

The increase in data collection throughout the past few years has aroused the interest of several researchers from different knowledge fields regarding data analysis. Consequently, data mining algorithms (DM) have been used to extract patterns for decision making. Amongst the tasks of DM, time-series classification (TSC) have been strongly highlighted. In this context, TSC, in respect to state-of-art algorithms and time-series representation techniques, consists on nontrivial tasks since domain and technique knowledge are required for their representation. The present paper presents a wide experimental evaluation of Shapelet Transform (ST) proposed by [Zalewski et al. \(2016\)](#), based on other classification models; besides, a new ST is presented aiming time-series representation, the extraction of significant attributes, and new data transformation with no significant losses by inferences. These techniques are based on the elimination of redundancies among the Shapelets, their correlation and self-correlation degree, as well as Shapelet ordination regarding any importance level induced in the proposed algorithms. After the separation of the Shapelet, predictive models were constructed based on distances, probabilities, search and optimization to show that the TS proposed by [Lines et al. \(2012\)](#) combined with some classic algorithms of attribute selection and aggregation, results in the reduction and simplification of the time-series complexity. Thus, any predictive model could be applied with no need for parameter adjustment; besides, superior accuracy performance could be achieved. We conducted our experiment in 25 clusters of data extracted from UCR repository ([Chen et al., 2015](#)) in order to evaluate the performance of the proposed model and methods. After the analysis of the obtained data, promising performances were observed from each TS used. **Keywords:**

classification, time serie, shapelet, data mining, machine learning.

Lista de Figuras

2.1	Representação ilustrativa das componentes de tendência e sazonalidade.	18
2.2	Representação da similaridade mediante duas palavras <i>Peter e Piotr</i>	19
2.3	Ilustração da diferença de alinhamento entre as distâncias Euclidiana e DTW. .	20
2.4	Resultado de agrupamento da transformada <i>shapelet</i> no conjunto de dados <i>Gun Point</i>	21
2.5	Ilustração de séries temporais para construção de modelos preditivos	24
2.6	Ilustração do impacto do algoritmo 1-NN	24
2.7	Impacto do algoritmo máquina de vetor de suporte linear (SVML), em conjunto de dados linearmente separável.	26
2.8	Impacto da rede neural artificial de múltiplas camadas na predição	27
2.9	Ilustração dos modelos preditivos baseados em procura	27
3.1	Ilustração da primitiva <i>shapelet</i>	29
3.2	Ilustração do processo de aplicação de uma janela deslizante.	30
3.3	Representação esquemática da localização da menor distância encontrada entre as subsequências <i>S</i> e <i>R</i>	32
3.4	Representação esquemática de uma linha de ordenação.	33
3.5	Ilustração do limiar de separação	34
4.1	Configuração de amostragens dos dados, uma adaptação de Faceli et al. (2011). .	39
4.2	Representação esquemática dos experimentos realizados	40
4.3	Variação quantitativa das melhores <i>k shapelet</i> selecionadas sobre os 25 conjuntos de dados mediante as extensões <i>TS+CFS</i> , <i>TS+CS</i> e <i>TS+PCA</i>	43
4.4	Diagrama de diferenças críticas das extensões mediante a seleção de <i>shapelet</i> .	43
5.1	Resultados de agrupamento do conjunto <i>Chlorine Conntrations</i>	46
5.2	Resultados de agrupamento do conjunto de dados <i>Gun Point</i>	46
5.3	Configuração das extensões nos modelos preditivos.	47
5.4	Diagrama de diferença crítica na <i>TS+CS</i>	50
5.5	Diagrama de diferença crítica para o pós-teste de <i>Naminye</i> na extensão <i>TS+PCA</i> . .	53
6.1	Variação do algoritmo <i>IBK</i> nas <i>ETS</i>	56
6.2	Diagrama de diferença crítica do algoritmo <i>IBK</i> nas <i>ETS</i>	56
6.3	Diferença crítica da variação do algoritmo <i>NB</i> mediante as extensões da <i>TS</i> . . .	57
6.4	Variação do algoritmo <i>Naive Bayes (NB)</i> nas extensões da <i>TS</i>	58
6.5	Diagrama de diferença crítica do algoritmo <i>SVML</i> nas extensões da <i>TS</i>	59
6.6	Variação do algoritmo <i>SVML</i> nas extensões da <i>TS</i>	60
6.7	Variação do algoritmo <i>Multilayer Perceptron (MLP)</i> nas extensões da <i>TS</i> . . .	61
6.8	Diferença crítica da variação (<i>MLP (TS+CFS)</i> , <i>MLP (TS+CS)</i> e <i>MLP (TS+PCA)</i>). .	61

6.9	Variação do algoritmo <i>Random Forest</i> nas extensões da TS.	63
6.10	Diagrama de diferença crítica do algoritmo Random Forest (RandFor) nas extensões	63
6.11	Variação do algoritmo J48	64
7.1	Comparação do algoritmo <i>SVML</i> na transformada TS(m/2) Versus o mesmo algoritmo na extensão da transformada <i>shapelet</i> <i>SVML</i> (TS+PCA).	68
7.2	Comparação do algoritmo <i>Random Forest</i> na transformada TS+(m/2) Versus <i>Random Forest</i> (TS+PCA).	68
7.3	Diagrama de diferença crítica para o pós-teste de <i>Nemenyi</i>	71
7.4	Resultados de acurácia dos tipos de séries temporais de acordo com as <i>Extensões</i> TS	72

Lista de Tabelas

4.1	Representação atributo-valor da transformada <i>shapelet</i>	36
4.2	Sumarização dos conjuntos de dados.	39
4.3	Valores utilizados no pós teste de Nemenyi e de Bonferroni-Dunn	41
4.4	Resultados das extensões da TS+ na buscas das melhores <i>shapelets</i>	42
5.1	Resultado experimental da avaliação da extensão TS+CFS em termo de acurácia	49
5.2	Resultado experimental da avaliação da extensão TS+CS em termo de acurácia	51
5.3	Resultado experimental da avaliação da extensão TS+PCA em termo de acurácia	52
6.1	Avaliação do algoritmo IBK nas extensões TS+	55
6.2	Avaliação do algoritmo Naive Bayes nas extensões TS+	57
6.3	Avaliação do algoritmo Máquinas de Vetores de Suportes Lineares (SVML) nas extensões TS+	59
6.4	Avaliação do algoritmo <i>Multilayer Perceptron</i> (MLP) nas extensões TS+	60
6.5	Resultados da variação do algoritmo Random Forest (RandFor) mediante as extensões TS+CFS, TS+CS e TS+PCA.	62
6.6	Avaliação do algoritmo J48 nas extensões	65
6.7	Avaliação comparativa dos melhores algoritmos em cada extensão da transformada	66
7.1	Avaliação comparativa da transformada TS em relação as extensões da TS . . .	69
7.2	Continuação da comparação da TS em relação a TS+	70
7.3	Demonstração dos resultados de acordo os tipos de séries temporais	72

Lista de Acrônimos

AM	Aprendizagem de Máquina
CFS	Correlation-base Feture Selection
CST	Classificação de Séries Temporais
CS	Consistency-base Filter
CD	Difrença Crítica
DTW	Dynamic Time Warping
ED	Distância Euclidiana
ETS	Extensão da Transformada Shapelet
MD	Mineração de Dados
MLP	Multilayer Perceptron
NB	Naive Bayes
PCA	Principal Component Analysi
PAA	Piecewise Agregate Aproximation
SAX	Symbolic Agrregate Approxima
SDA	Shape Description Alphabet
SVML	Máquina Suportes de Vetores Lineares
TS	Transformada Shapelet
KNN	K-Nearest Neighbors

Outras Anotações

<i>acc</i>	Acurácia
<i>T</i>	Conjunto de todas séries temporais
<i>C</i>	Conjunto de todas as classes
<i>min</i>	Tamanho mínimo de subsequências
<i>max</i>	Tamanho máximo de subsequências
<i>L</i>	Limiar de separação
<i>VP</i>	Verdadeiros positivos
<i>VN</i>	Verdadeiros negativos

Sumário

1	Introdução	14
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.2	Organização do Trabalho	16
2	Mineração de Dados em Séries Temporais	17
2.1	Considerações Iniciais	17
2.1.1	Fundamentos e Análise de Séries Temporais	17
2.1.2	Medidas de Similaridades	19
2.1.3	Medidas baseadas na forma	20
2.2	Aprendizado de Máquinas	21
2.2.1	Agrupamento em Séries Temporais	21
2.2.2	Classificação em Séries Temporais	22
2.2.3	Redução da Dimensionalidade	22
2.3	Construção de Modelos Preditivos	23
2.3.1	Modelos Preditivos Baseados em Distâncias	24
2.3.2	Modelos Preditivos Baseados em Probabilidade	25
2.3.3	Modelos Preditivos Baseados em Otimização	25
2.3.4	Modelos Preditivos Baseados em Procura	26
2.4	Considerações Finais	27
3	Classificação Por Meio das <i>Shapelets</i>	28
3.1	Considerações Iniciais	28
3.1.1	Gerar Subsequências Candidatas	30
3.1.2	Calcular a Distância	31
3.1.3	Determinar a Qualidade	32
3.2	Considerações Finais	34
4	Explorando a Transformada <i>Shapelet</i>	35
4.1	Considerações Iniciais	35
4.2	Transformada <i>Shapelet</i>	35
4.2.1	Abordagem Reduzida <i>TS+</i>	36
4.3	Abordagens da Transformada	37
4.4	Criação das Reduzidas <i>TS</i>	37
4.5	Configuração e Descrição dos Dados	38
4.6	Método Holdout	38
4.6.1	Configuração dos Experimentos	38
4.7	Análises Estatísticas	41

4.8	Avaliação de Resultados das ETS	42
4.9	Considerações Finais	44
5	Avaliação Experimental das ETS	45
5.1	Considerações Iniciais	45
5.2	Resultados de Agrupamentos	45
5.3	Configuração dos Modelos Preditivos	46
5.4	Avaliação	48
5.4.1	Avaliação da Extensão <i>TS+CFS</i>	48
5.4.2	Avaliação da Extensão <i>TS+CS</i>	50
5.4.3	Avaliação da Extensão <i>TS+PCA</i>	50
5.5	Considerações Finais	53
6	Avaliação dos Algoritmos Preditivos	54
6.1	Considerações Iniciais	54
6.2	Avaliação do Algoritmo <i>IBK</i>	54
6.3	Avaliação do Algoritmo <i>Naive Bayes (NB)</i>	56
6.4	Avaliação do Algoritmo <i>SVM</i>	58
6.5	Avaliação do Algoritmo <i>MLP</i>	58
6.6	Avaliação do Algoritmo <i>Random Forest</i>	61
6.7	Avaliação do Algoritmo <i>J48</i>	63
6.8	Comparação dos Melhores Algoritmos	63
6.9	Considerações Finais	64
7	Comparação da TS Versus as ETS	67
7.1	Considerações Iniciais	67
7.2	Avaliação de Resultados	67
7.3	Considerações Finais	71
8	Conclusão	73
8.1	Limitações	73
8.2	Trabalhos Futuros	74
	Referências Bibliográficas	75

Capítulo 1

Introdução

A classificação de séries temporais (CST) tem sido destacada na literatura como uma das tarefas de mineração de dados com elevado crescimento de estudos, devido aos grandes volumes de dados que a cada dia aplicações de diversos domínios de conhecimentos têm gerado. Neste contexto, resume-se que qualquer dado coletado de modo sequencial pode ser representado mediante uma série temporal. Uma série temporal pode ser compreendida como um conjunto ordenado de observações registradas cronologicamente (Morettin e Toloi, 2006). Com o propósito de utilizar métodos específicos para análises e comportamentos contidos nas séries temporais, abordagens baseadas em técnicas de mineração de dados têm sido exploradas nas últimas décadas, como é fundamentado nos trabalhos de Antunes e Oliveira (2001); Morchen (2003) e Zalewski et al. (2016). Como mencionado anteriormente, dentre as tarefas de mineração de dados, a classificação de séries temporais tem sido amplamente explorada, consequentemente algoritmos de classificação foram desenvolvidos para considerar a dependência temporal dos dados (Cotofrei e Stoffel, 2002; Xing et al., 2010), ao contrário dos algoritmos tradicionais que não consideram a existência de relação temporal presente nos dados (Bagnall, 2012).

A maior parte dos métodos propostos na literatura de séries temporais para construção de classificadores é baseada em uma representação de características estatísticas ou em ajustes de parâmetros, como é descrito em Bakshi et al. (1994) e Cotofrei e Stoffel (2005). Esta representação pode influenciar de modo negativo na compreensão do conhecimento dos modelos, pois, encontra-se distante da intuição humana, a qual baseia-se principalmente na morfologia da série. Para contornar estes problemas, a utilização da primitiva *shapelet*, proposta por Ye e Keogh (2009), é fortemente recomendada por ser um descritor de características morfológicas de séries temporais que permite melhorar a compressão dos conceitos ou da simbologia presente, devido a sua maior proximidade com a percepção humana, promovendo a descoberta de padrões em séries temporais (Ye e Keogh, 2009) e (Zalewski et al., 2016). A CST através da primitiva *shapelet* pode ser dividida em duas abordagens, a primeira é chamada de abordagem embutida, pois, o processo de identificação de *shapelet* ocorre internamente no modelo preditivo, enquanto que na segunda abordagem, o processo de identificação de *shapelet* ocorre separadamente em relação à fase de construção dos modelos preditivos, chamada de abordagem transformada (Bagnall, 2012) ou transformada *shapelet* (TS). A TS é composta de um número elevado de subsequências ou *shapelet* que formam atributo-valor, porém nem todas elas são utilizadas para inferências. Neste contexto, pesquisamos como a transformada *shapelet* em combinação com os algoritmos de seleção de atributos pode ser melhor utilizada para encontrar as melhores *shapelets* com características discriminatórias e reduzir de modo significativo o tamanho e a representação das séries temporais. Portanto, esta pesquisa é uma continuação ou extensão dos trabalhos de Hills et al. (2014) e de Zalewski et al. (2016). Nossas contribuições são descritas da seguinte maneira:

1. Exploramos as extensões TS+CFS e TS+CS proposta por [Zalewski et al. \(2016\)](#), por meio de outros algoritmos de classificação não testados anteriormente.
2. Testemos um novo algoritmos de seleção e agregação de atributos (PCA), com o propósito de selecionar as melhores *shapelets* com características discriminatórias em séries temporais.
3. Comparamos os resultados das melhores k *shapelet* selecionadas com os resultados do método de [Hills et al. \(2014\)](#) em termos quantitativos de *shapelet* selecionadas que formam atributo-valor.
4. Realizamos um estudo sobre a classificação de séries temporais por meio de algoritmos de diferentes paradigmas, ao contrário do trabalho de [Zalewski et al. \(2016\)](#) no qual o autor destaca somente a classificação de séries temporais baseada em árvore de decisão. Os modelos adotados nesta pesquisa foram utilizados sem a necessidade de estimativas de parâmetros ou ajustes de funções.
5. Apresentamos uma proposta baseada na aplicação de uma técnica não utilizada dentro da abordagem da transformada *shapelet*, bem como nos dados brutos temporais, com o propósito de atingir os objetivos preconizados nesta pesquisa.

1.1 Objetivos

Fundamentando o cenário descrito sobre a classificação de séries temporais de acordo com a transformada *shapelet*, os objetivos desta dissertação são descritos da seguinte maneira:

1.1.1 Objetivo Geral

Melhorar o modo de seleção das melhores *shapelets* compostas de características discriminatórias por meio de algoritmos de seleção e agregação de atributos.

1.1.2 Objetivos Específicos

Especificam-se da seguinte maneira os objetivos:

1. Testar algoritmos clássicos de seleção de atributos para selecionar as *shapelets* de acordo algum critério de avaliação entre elas.
2. Propor um método de construção de algoritmos de predição de séries temporais baseados em modelos de inferências diferentes.
3. Demonstrar por meio das especificações 1 e 2, que as extensões da transformada *shapelet* (ETS) por meio de seus resultados, possibilitam o aumento de desempenho dos classificadores em termos de acurácia, sem a necessidade de nenhum ajuste de parâmetro complexo.

Definidas as seguinte hipótese:

- A qualidade de representação das extensões da transformada *shapelet* possibilita a construção de modelos baseados em distância, probabilidade, procura e otimização.

Baseados nas seguintes assertivas:

- Em problemas que envolvem a extração de características morfológicas, a transformação das séries temporais em outro domínio de representação pode prover melhor desempenho em termos de acurácia em relação a outras abordagens baseadas no domínio do tempo (Lines et al., 2012).
- A existência e a combinação de múltiplas *shapelets* é fundamental para a classificação de séries temporais (Hills et al., 2014).
- A separação do processo de identificação das *shapelets* em relação a fase de indução dos modelos de predição possibilita a adaptação de novas abordagens inerentes à mineração de dados.

1.2 Organização do Trabalho

Este trabalho está estruturado pelos seguintes capítulos:

1. Mineração de Dados:

Este capítulo apresenta os fundamentos sobre mineração de dados em séries temporais, suas representações e uma descrição resumida sobre aprendizagem de máquina, sua utilização e a construção dos modelos preditivos.

2. Classificação Mediante Shapelets:

Neste capítulo enfatizamos a classificação de séries temporais utilizando a primitiva *shapelet*, e destacamos os principais algoritmos utilizados para identificação das *shapelets*.

3. Explorando a Transformada Shapelet:

Neste capítulo são apresentadas as propostas deste trabalho, cujo objetivo é explorar a transformada *shapelet* (TS) e os principais métodos utilizados para criar cada extensão da TS.

4. Análises Estatísticas:

O capítulo destaca as principais técnicas de análises estatísticas com o propósito de avaliar os desempenhos dos modelos criados por meio das diferenças estatísticas existentes entre os mesmos.

5. Comparação dos Algoritmos:

Neste capítulo é apresentada uma comparação dos algoritmos preditivos de acordo com os resultados de acurácia como medida de desempenho, com base nas variações das extensões da transformada propostas.

6. Comparação da TS versus as ETS:

Neste capítulo é feita uma comparação dos resultados da transformada *shapelet* (TS) em relação as suas extensões.

7. Conclusão:

Neste capítulo são apresentadas as principais contribuições desta pesquisa bem como as limitações constatadas e os futuros trabalhos a serem desenvolvidos.

Capítulo 2

Mineração de Dados em Séries Temporais

2.1 Considerações Iniciais

Ao longo dos anos as análises de séries temporais têm sido aplicadas em diversos ramos do conhecimento, como por exemplo na economia (a análise da variação cambial, a redução ou o aumento do preço das ações, o aumento ou declínio da produção industrial, etc..) ou na meteorologia (o estudo das precipitações atmosféricas, análise de previsão do tempo bem como a medição da velocidade do vento) (Ehlers, 2005).

Neste cenário, várias informações surgem para contribuir na formação de base de dados em grandes escalas de modo sequencial (Zalewski et al., 2016). Se esses dados são coletados ao longo do tempo e apresentam uma variação temporal, então podem ser representados por intermédio de uma série temporal (Morettin e Toloi, 2006). A literatura destaca com maior ênfase, a respeito dos métodos clássicos para análise dos eventos contidos em séries temporais, que tais tarefas não são triviais e dependem do domínio da aplicação (Zalewski, 2015). Nesse contexto, são recomendados os modelos baseados em mineração de dados (MD) para a análise de séries temporais (Morchen, 2006).

Os processos de MD podem ser resumidos da seguinte maneira:

- **Pré-processamento:**

O desempenho dos algoritmos adotados em MD pode ser afetado pelo estado dos dados. A tarefa de pré-processamento consiste num processo de melhorar a qualidade dos dados mediante a eliminação de ruídos (Faceli et al., 2011). Nesse processo destacam-se ainda as tarefas: integração dos dados, limpeza e a redução dos dados mediante a seleção de atributos e de redução de exemplos.

- **Extração de Conhecimento:**

Esta atividade visa à busca de padrões que permitem representar o conhecimento de uma maneira consistente em um determinado conjunto de dados.

- **Pós-processamento:**

Este processo consiste na avaliação e validação das etapas anteriores. Normalmente essa etapa faz uso de técnicas de visualização, estatísticas e inteligência artificial.

2.1.1 Fundamentos e Análise de Séries Temporais

Como mencionado anteriormente, uma série temporal consiste em um conjunto ordenado de observações de um determinado fenômeno realizado de modo sequencial ao longo do

tempo, não necessariamente igualmente espaçadas (Ehlers, 2005). Neste contexto são notadas características fundamentais que se baseiam na dependência entre as observações (instante de tempo). Formalmente, uma série temporal é definida como:

Definição 2.1 (Série Temporal). *Uma série temporal $T = \{t_1, \dots, t_i, t_j, \dots, t_m\}$ consiste em um conjunto de m valores ordenados, $m \geq 2$, tal que se $i < j$, t_i ocorre cronologicamente antes que t_j .*

Portanto, uma série temporal pode ser interpretada em função das subsequências que a compõem, nesse contexto, uma subsequência é definida como:

Definição 2.2 (Subsequência). *Uma subsequência $S = \{t_p, \dots, t_{p+n-1}\}$ consiste em um subconjunto contíguo de n valores de T com início na posição p , tal que $2 \leq n \leq m$ e $1 \leq p \leq m - n + 1$.*

Na compreensão dos eventos que as séries temporais apresentam, a literatura destaca técnicas ou conceitos de suas decomposições dentro de um conjunto finito de componentes independentes. As principais componentes são: *tendências*, *sazonalidade* e *resíduo* (Pyle, 1999; Morettin e Toloi, 2006).

Nesse contexto, cada uma das observações t_i que compõem uma determinada série temporal T , podem estar influenciadas por uma ou mais dessas componentes. Portanto, em grande parte dos problemas estudados, não é possível identificar diretamente a atuação dessas componentes na série temporal, de modo que somente podem ser extraídas e compreendidas por meio da aplicação de técnicas específicas de decomposição (Brockwell e Davis, 1996). Uma vez que as componentes *sazonal* e *tendência* possuem uma relação forte, e uma delas pode afetar os métodos de análises, é possível separar uma da outra (Brockwell e Davis, 1996; Morettin e Toloi, 2006).

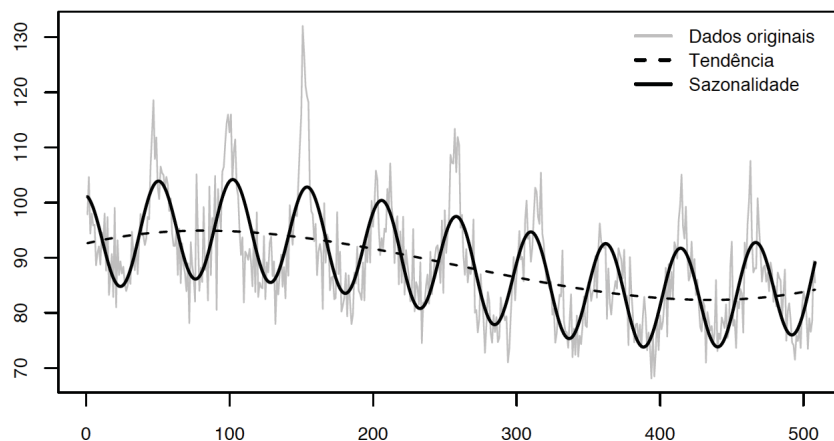


Figura 2.1: Representação ilustrativa das componentes de tendência e sazonalidade (Ferrero, 2009).

De um modo resumido são destacadas as definições e fundamentos das componentes acima mencionadas:

1. **Tendência:** o movimento dominante em uma série temporal é chamado de tendência, ela exerce influência nas observações por longos instantes, chega a alterar o nível médio da série. Na figura 1 a tendência é representada pela linha tracejada, nesse caso a série do fenômeno é observada em cor cinza. Ainda nesse contexto, as séries apresentam comportamentos diferentes de tendência onde estão baseados os métodos de identificação da mesma componente (EHLERS, 2005).

2. **Sazonalidade:** a sazonalidade é um comportamento que se repete em diferentes instantes de acordo com alguma característica, ela apresenta oscilação ao longo da tendência (Ferrero, 2009). A identificação desta componente é um processo importante no que tange a análise de séries temporais, a sua presença permite a descoberta de informações relevantes, enquanto que na sua ausência a remoção pode afetar significativamente outras características da série (Maletzke et al., 2009). Na figura 1 a componente é representada por uma linha contínua de cor preta.
3. **Resíduo:** ainda na análise de séries temporais, o fenômeno constatado pode apresentar eventos aleatórios de características inerentes ao mesmo fenômeno (Maletzke et al., 2009). Nesse contexto, estes comportamentos são representados pela componente resíduo. Portanto, a modelagem das duas componentes mencionadas anteriormente permitem isolar e analisar a componente resíduo (Ehlers, 2005).

2.1.2 Medidas de Similaridades

O conceito de similaridade tem sido fundamental para análises de séries temporais especialmente nas tarefas de mineração de dados (Zalewski, 2015). Uma medida de similaridade pode ser como:

$$Dist : T_x \times T_y \rightarrow \mathbb{R} \quad (2.1)$$

Em muitos casos, o conceito de similaridade é possível ser ilustrar por meio de algumas transformações dos objetos em causa. Por exemplo, a figura 2.2 ilustra o quão duas palavras *Peter* e *Piotr* são semelhantes.

Nesse contexto é necessário estabelecer alguns parâmetros de custos: *substituição*, *inserção* e *exclusão de uma unidade*.

- De *Peter* para *Piter* (substituição de *i* para *e*)
- De *Piter* para *Pioter* (inserção de *o*).
- De *Pioter* para *Piotr* (exclusão de *e*).

Podemos enfatizar que esta transformação leva em consideração que o custo a ser aplicado seja menos trivial para que este processo seja completado.

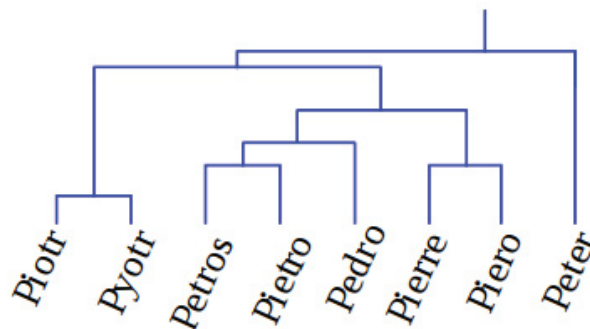


Figura 2.2: Representação da similaridade mediante duas palavras *Peter* e *Piotr*.

De acordo com a literatura as medidas de similaridades de séries temporais podem ser divididas em quatro categorias principais (Morchen, 2006; Esling e Agon, 2012): abordagens baseadas na forma, características, edição e baseadas na estrutura. Dentre as categorias, nosso trabalho enfatiza com maior descrição a medida baseada na forma.

2.1.3 Medidas baseadas na forma

O objetivo principal destas medidas é a análise morfológica de séries temporais (Castro, 2012). Normalmente a literatura recomenda o uso das medidas baseadas na forma, particularmente a *distância euclidiana (ED)* (Keogh e Kasetty, 2003), devido às limitações que essas medidas apresentam no que tange aos momentos de amplitudes (escalas e translação), também é recomendada a medida *dynamic time warping (DTW)* (Sakoe e Chiba, 1978), com o propósito de combinação das duas medidas (ED e DTW) para resolver a invariância da complexidade das séries temporais.

Distância Euclidiana

De acordo com os estudos de Zalewski (2015), o autor destaca a distância euclidiana como sendo uma medida que determina a distância adequada existente em linha reta entre dois pontos no mesmo espaço dimensional (m). O cálculo da distância Euclidiana $Dist_{ED}$ entre duas séries temporais T_x e T_y é definida conforme a Equação 2.2:

$$Dist_{ED}(T_x, T_y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (2.2)$$

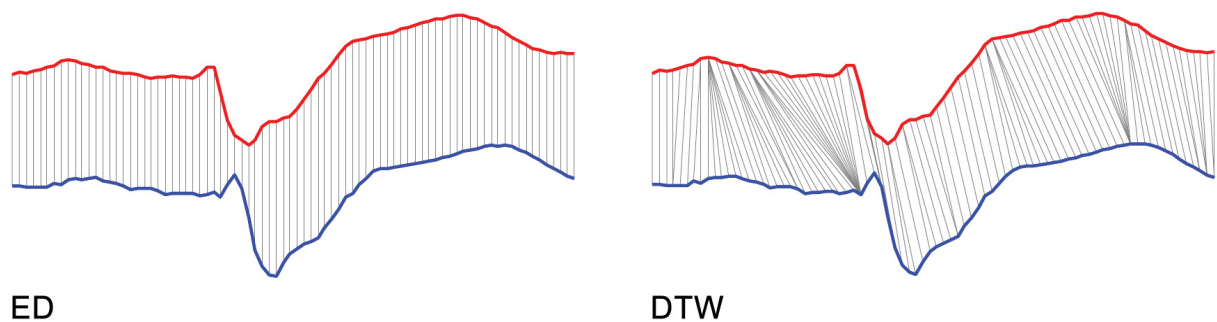
onde i indica cada dimensão em cada iteração para o cálculo da distância da série.

Dynamic Time Warping

De modo resumido, a distância DTW permite o alinhamento entre duas séries temporais, tratando de possíveis distorções ou de tamanhos diferentes.

Na figura 2.3 é representada a diferença de alinhamento entre as medidas de similaridade ED e DTW.

Figura 2.3: Ilustração da diferença de alinhamento entre as distâncias Euclidiana e DTW (Keogh et al., 2003) e (Silva et al., 2013)



Nos trabalhos recentes de DING (2008) e WANG (2013) os autores constataram por meio de uma extensa avaliação empírica, que os algoritmos de classificação baseados em 1-vizinho mais próximo (1-NN) representada pela distância Euclidiana e pela medida DTW apresentam bons resultados em termo de acurácia em vários problemas de domínio de séries temporais.

Uma das justificativas para este comportamento é apresentada em Hills et al. (2014), nesse estudo, os resultados experimentais demonstraram que ED é a medida elástica, enquanto que DTW é a medida com distorção no tempo. Como dito anteriormente a distância DTW permite o alinhamento entre duas séries temporais, possibilitando o tratamento de possíveis distorções, enquanto que a ED determina qual é a distância em uma linha reta entre dois pontos

que pertencem ao mesmo espaço dimensional (Zalewski, 2015). Os algoritmos baseados nessas abordagens têm sido destacados na literatura de séries temporais como estado-da-arte (DING, 2008), (Batista et al., 2011) e (WANG, 2013). A maior parte dos métodos propostos na literatura de séries temporais para construção de classificadores são baseados em uma representação de características estatísticas ou em ajustes de parâmetros, como é descrito em Bakshi et al. (1994) e Cotofrei e Stoffel (2005).

2.2 Aprendizado de Máquinas

As técnicas de aprendizagem de máquina (AM) propostas na literatura, contornam limitações dos modelos estatísticos para análise de séries temporais (Zalewski, 2015). Dentre as tarefas de AM são destacadas duas tarefas principais: agrupamento e classificação, sendo a classificação o foco deste trabalho. Em aprendizagem de máquina são diferenciados três tipos de aprendizagens, aprendizado supervisionado, não supervisionado e aprendizado por reforço, e nestes estudos destacamos apenas os dois primeiros.

No aprendizado supervisionado, o algoritmo de indução é aplicado ao conjunto de dados de treinamento, nesse contexto a classificação de cada exemplo é mediante a classe que é conhecida. Caso seja um conjunto de valores ordenados, o algoritmo resulta em um classificador. Se for um conjunto de valores ordenados e infinitos o algoritmo gera um regressor. Na regressão, dado um exemplo sem a classe, o algoritmo atribui uma possível classe conhecida.

No âmbito de aprendizado não supervisionado, a classe no conjunto de dados de treinamento não é conhecida. Uma das técnicas utilizadas é o agrupamento, tendo como objetivo encontrar grupos de exemplos similares mediante alguma medida de similaridade aplicada em suas características (Faceli et al., 2011).

2.2.1 Agrupamento em Séries Temporais

A tarefa de agrupamento de séries temporais consiste em associar um conjunto de dados no mesmo espaço dimensional. Nesse contexto, a similaridade entre as séries pertencentes à mesma classe é maximizada, enquanto que a similaridade entre as séries de classes diferentes é minimizada (Zalewski, 2015), ou seja, os exemplos similares tendem a estar mais próximos uns dos outros. Os métodos tradicionais de agrupamentos podem ser aplicados como descrito em Han e Kamber (2006), e uma revisão destes métodos está disponível em Liao (2005); Rani e Sikka (2012).

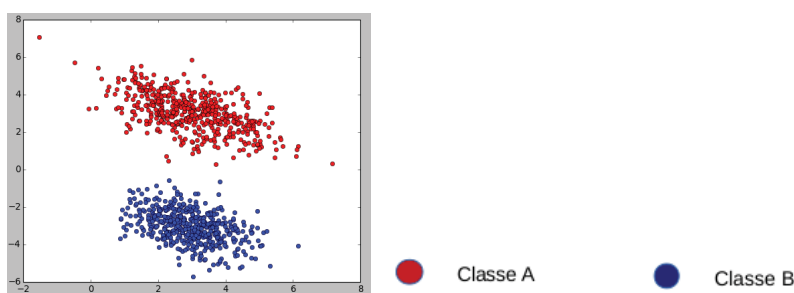


Figura 2.4: Resultado de agrupamento da transformada *shapelet* no conjunto de dados *Gun Point*.

Através de um problema de duas classes a figura 2.4 ilustra um agrupamento feito nos dados transformados pelos métodos empregados neste trabalho antes da classificação.

2.2.2 Classificação em Séries Temporais

A CST por meio dos métodos propostos na literatura é fortemente ligada ao tipo de representação utilizado (Zalewski, 2015). Neste contexto, têm sido destacadas duas abordagens, a primeira corresponde à utilização dos dados originais surgindo a necessidade de adaptação dos algoritmos clássicos de aprendizagem de acordo com o problema. Na segunda abordagem, as séries temporais são transformadas em uma representação de características, tais características são utilizadas para indução de classificadores de aprendizagem de máquina.

Portanto, a classificação consiste em determinar uma função que permite associar uma classe a uma série temporal que não esteja rotulada, ou seja, sem a classe. Nesse contexto, a classificação pode ser formalizada da seguinte forma; seja \mathbb{T} o conjunto de todas as séries possíveis de um determinado domínio e seja $C = \{c_1, \dots, c_w\}$ um conjunto de w classes, tal que:

$$\forall T_i \in \mathbb{T} : ((T_i \in c_1) \vee \dots \vee (T_i \in c_w)) \wedge (T_i \in c_j \rightarrow T_i \notin c_k, j \neq k) \quad (2.3)$$

um classificador de séries temporais consiste em uma função f que permite mapear uma série $T_i \in \mathbb{T}$ para uma classe $c \in C$:

$$f : \mathbb{T} \rightarrow \{c_1, \dots, c_w\} \quad (2.4)$$

2.2.3 Redução da Dimensionalidade

Ao filtrar atributos redundantes de acordo com algum critério de importância, tal processo não requer nenhuma interação direta do algoritmo de aprendizado. A medida de importância baseia-se unicamente nas propriedades intrínsecas dos dados que estão relacionados aos atributos. Ao efetuarmos a filtragem de atributos e reduzirmos o conjunto de dados em pequena quantidade, cumpre-se o objetivo da redução da dimensionalidade dos dados, sendo um resultado significativo nas tarefas de MD.

A redução da dimensionalidade em séries temporais pode ser tratada em aprendizagem de máquinas através de algoritmos de seleção e agregação de atributos ou outras medidas de buscas de atributos com maior ganho de informação (Faceli et al., 2011). Neste contexto, filtrar atributos irrelevantes ou redundantes mediante algum critério de avaliação pode ser tarefa do algoritmo específico de seleção, buscando atributos que contêm características discriminantes na série temporal antes da construção dos modelos de predição.

Portanto, na avaliação dos experimentos feitos neste estudo, utilizamos os seguintes algoritmos baseados nas abordagens de seleção de atributos: *Consistency-based Filter*(CS) Liu et al. (1996), *Correlation-based Feature Selection* (CFS), e um método de agregação *Principal Component Analysis* (PCA) (Smith et al., 2002). Esses métodos de seleção de subconjunto de atributos são frequentemente usados em diversos problemas em que for constatada a maldição da dimensionalidade.

Consistency-based Filter-(CS)

CS: é um algoritmo que funciona através de uma heurística probabilística, selecionando um subconjunto de poucos atributos que apresentam uma veracidade maior com os atributos classes. Segundo Zalewski et al. (2016) essa abordagem detém um ganho de informação na combinação de atributos que permite dividir os dados em pequenos subconjuntos com uma classe majoritária, onde sua complexidade de tempo no pior caso é $O(MK^2)$. O fator de busca por uma

direção específica não altera o tempo de execução. Nesse contexto M representa a quantidade de dados que formam as séries temporais, e K o número de atributos, ou seja, são as *shapelets* selecionadas.

Correlation-based Feature Selection-(CFS)

CFS: sua heurística baseia-se em um modelo de avaliação de subconjuntos de atributos mediante um critério de medição de atributos com maior grau de relevância daqueles que apresentam redundâncias entre si. Esse algoritmo avalia um subconjunto de atributos levando em conta a capacidade de predição individual de cada um e a redundância entre eles (Zalewski et al., 2016). Para maior esclarecimento desta abordagem, basta lembrar de uma das medidas clássicas de correlação que é a de *Pearson*, que consiste na divisão da covariância das variáveis pelos produtos de cada desvio padrão obtido, isto é, a dispersão dos valores em torno da média (Figueiredo Filho e Junior, 2010). Nesse contexto, é necessário que as variáveis sejam quantitativas e quanto a sua complexidade de tempo no pior caso também é de $O(MK^2)$.

Principal Component Analysis-(PCA)

PCA: é uma das principais técnicas de agregação usada para redução de dados. Sendo um modelo estatístico para correlacionar os exemplos e eliminar a redundância (Pearson, 1901) e (Hotelling, 1933). Assim, é feita uma combinação de atributos originais através de uma função que seja linear ou não (Faceli et al., 2011). Apesar das técnicas de agregação em muitos casos levarem a perda de informações relevantes de valores originais, foi necessário trazer este modelo neste experimento para provar nossa abordagem num sentido mais amplo e com o propósito de obter melhores resultados em termo de redução de atributos, bem como nos resultados de acurácia dos modelos preditivos usados nesta avaliação.

Segundo os resultados obtidos pela transformada shapelet (*TS*) e os algoritmos de seleção de subconjuntos de atributos descritos anteriormente, propomos três algoritmos a serem avaliados, *TS+CFS*, *TS+CS*, *TS+PCA*. Nessa abordagem de combinações foi realizada uma busca baseada no algoritmo *Best First* e a direção de busca adotada é a *forward*, essa configuração é fundamentada no trabalho de (Zalewski et al., 2016).

2.3 Construção de Modelos Preditivos

Em aprendizado de máquina existem vários algoritmos com a finalidade de descoberta de padrões em conjuntos de dados para inferências ou predição. Um modelo preditivo pode ser entendido como uma função que dado um conjunto de dados já rotulados (contendo a classe) constrói um estimador (Faceli et al., 2011).

Como dito anteriormente, neste estudo foram construídos modelos preditivos baseados em *distância*, *probabilidade*, *otimização* e *procura*, com o propósito de explorar a diversidade aleatória de inferências de algoritmos diferentes.

Como exemplo da construção de séries temporais, é destacada a simulação de dois atores que seguram uma réplica de uma arma e apontam para o alvo com duração de *um segundo*, posteriormente colocam a arma novamente no coldre. A ideia foi mapear a semelhança existente entre os dois atores naqueles instantes, a figura 2.5 demonstra apenas o eixo x que representa o movimento de um dos atores, através desta figura vamos demonstrar o comportamento dos modelos preditivos mencionados anteriormente para avaliação das extensões criadas.

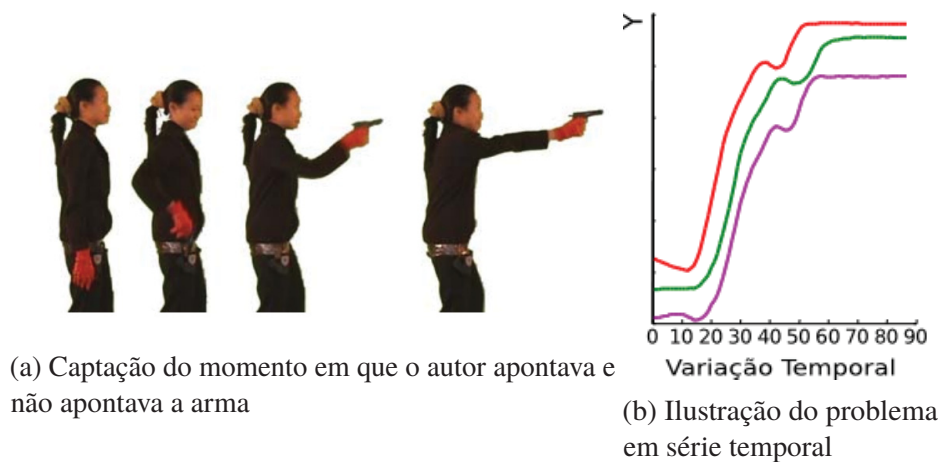


Figura 2.5: Ilustração do problema do conjunto de dados temporais (Gun/Point) baseado em [Keogh et al. \(2003\)](#). A figura 2.5(a) ilustra os movimentos que o ator realiza até que aponte a arma ao alvo. Enquanto que a figura 2.5(b) ilustra a conversão destes movimentos em séries temporais.

2.3.1 Modelos Preditivos Baseados em Distâncias

Durante vários anos os modelos de classificação baseados em distâncias foram os primeiros a serem destacados nas tarefas de classificação de séries temporais. E hoje ainda são considerados estado-da-arte, em que suas técnicas se resumem na aproximação existente entre os exemplos, isto é, dados similares tendem a estar aglomerados na mesma região ([Faceli et al., 2011](#)). Uma das medidas utilizadas é a distância euclidiana e dentro desta abordagem destacamos o algoritmo 1-Vizinho mais próximo (*1-Nearest Neighbor*) que consequentemente é uma das técnicas mais usadas dentro desta abordagem ([Batista et al., 2011](#)). Embora o algoritmo 1NN apresente características de um algoritmo preguiçoso, do inglês (*lazy*), em contrapartida contempla um aprendizado interessante neste estudo, pois sua abstração de aprendizado e predição de cada série temporal mediante nossa proposta, expande nossa visão nos mais variados tipos de predição que podem ser realizadas dentro desta abordagem.

Dentro de um ambiente supervisionado, o algoritmo em fase de seu treinamento memoriza os exemplos rotulados pertencentes ao conjunto de treinamento, posteriormente o rótulo da classe que foi associado nos dados de treinamento mais próximo do conjunto de teste é usado para classificar novos exemplos ([Faceli et al., 2011](#)).

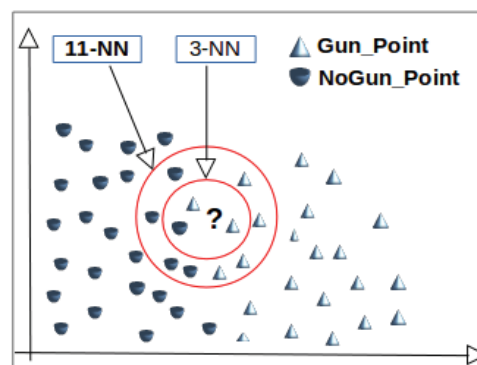


Figura 2.6: Ilustração do algoritmo 1-NN seu impacto no valor de K, num conjunto de dados temporais (Gun Point).

A equação 2.3 representa a distância euclidiana sendo a métrica mais utilizada pelo algoritmo 1NN. Esse é um classificador não paramétrico, a figura 2.6 representa um problema de duas classes, dado um exemplo ("??") não rotulado, encontrar os K vizinhos mais similares na base rotulada (treinamento), e atribua a classe mais frequente para "?". Neste cenário K representa o número de exemplos mensurados na base de treinamento, normalmente tem sido utilizada distância euclidiana.

Na seção 2.1.3 discutiu-se mais detalhadamente essa medida, bem como outras usadas para construção de modelos preditivos baseados em distâncias.

2.3.2 Modelos Preditivos Baseados em Probabilidade

O aprendizado baseado em probabilidades ou teoria de *Bayer* ou ainda redes Bayesianas normalmente são aplicados em casos de incertezas em que as informações disponíveis para análises são incompletas e imprecisas (Faceli et al., 2011), sua aplicabilidade é notada nas áreas de processamento de imagem, bioinformática, engenharia e medicina. E nesse estudo não seria diferente aplicar a mesma teoria em nossos problemas. Um dos objetivos e motivação do uso do mesmo modelo é pelo fato de fornecer de uma maneira clara sobre o cálculo da probabilidade *a posterior* de um evento, mediante os dados de suas probabilidades *a priori* (Faceli et al., 2011) bem como a harmonia de um novo dado. Para a classificação, estes modelos normalmente têm utilizado o algoritmo *Naive Bayes* onde seu termo *naive* deriva da teoria de que os atributos dos dados de treinamento ou testes são independentes de sua classe (Faceli et al., 2011), ou seja, não existe uma relação direta dos atributos e a classe predita.

Destacando a figura 2.5 demonstramos um problema de duas classes. No modo probabilístico retratamos a solução pelo modelo de classificação da seguinte maneira:

- *GunPoint*: Representa a classe do movimento que os atores fazem quando apontam para o alvo.
- *NoGunPoint*: Representa a classe onde os atores não estão apontando a arma para o alvo.

$$P(GunPoint|NoGunPoin) = \frac{P(GunPoint|NoGunPoin)P(GunPoint)}{P(NoGunPoint)} \quad (2.5)$$

onde $P(GunPoint|NoGunPoin)$ demonstra qual é a probabilidade de ser a classe onde os atores não estão apontando para o alvo, dado que a indução ao modelo é do instante em que eles estão apontando para o alvo.

$P(GunPoint)$ representa qual é a probabilidade de ser a classe que os atores estão apontando para o alvo.

$P(NoGunPoint)$ determina qual é a probabilidade de ser a classe onde os autores não estão apontando para o alvo.

Por meio da equação 2.5 demonstramos o fundamento inicial de como ocorre a classificação baseada em probabilidade.

2.3.3 Modelos Preditivos Baseados em Otimização

No uso dos modelos baseados em otimização destacam-se as máquinas de vetores de suporte (SVMs), sendo a teoria de aprendizados estatísticos proposto por (Vapnik, 1995) e as redes neurais artificiais-RNAs (Braga 2007, Haykin, 1991). Suas metodologias praticamente consistem

na otimização de alguma função, em que o problema de aprendizado para predição consiste em minimizar ou maximizar uma função objetivo (Faceli et al., 2011). Para avaliação dessas abordagens fez-se necessário o uso dos classificadores baseados em Máquinas de Vetores de Suportes Lineares (SVML) e os baseados em redes neurais artificiais *Multilayer Perceptron* (MLP), porque já existem resultados na literatura com estes modelos por meio de outras abordagens de CST, e podem ser comparados nos experimentos desenvolvidos nesta pesquisa.

A figura 2.7 ilustra em sua primeira fase o problema de duas classes já mencionadas anteriormente com o intuito de ser resolvido linearmente pelo modelo SVML, pelas regras de funções e otimizações, na segunda fase, de acordo com a figura 2.7 nota-se que o modelo cria vetores de suportes (exemplos próximos ao hiperplano ótimo) que maximizam a distância entre as duas classes e minimizam a distância intra-classes (exemplos da mesma classe), possibilitando a resolução linear do problema.

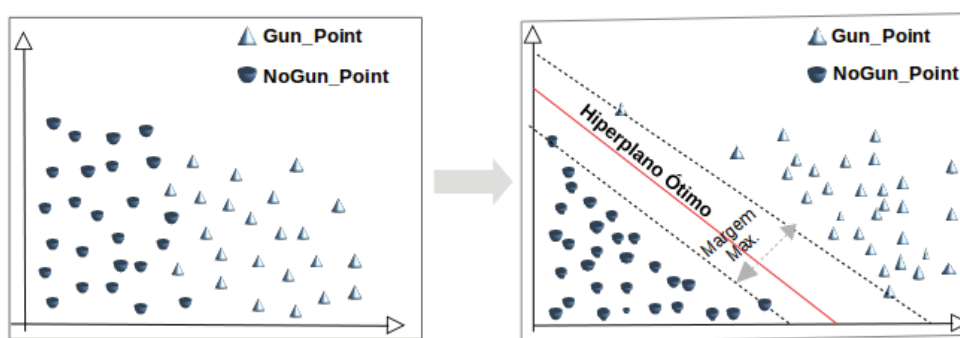


Figura 2.7: Impacto do algoritmo máquina de vetor de suporte linear (SVML), em conjunto de dados linearmente separável.

Ainda na construção dos modelos preditivos baseados em otimização, a figura 2.8 destaca o mesmo problema de um conjunto de dados de duas classes. Com auxílio de redes neurais artificiais, especificamente a rede MLP com algoritmo de treinamento *Backpropagations*, o problema é resolvido com base numa configuração de duas camadas escondidas, sendo cinco neurônios na primeira camada, três neurônios na segunda camada e finalmente dois neurônios de saída. Na primeira camada o modelo aprende sobre os exemplos e na segunda camada após certas interações o modelo já cria certas divisões, finalmente têm-se como a saída pelos dois neurônios, a predição da classe em cada exemplo.

2.3.4 Modelos Preditivos Baseados em Procura

A classificação baseada em procura normalmente tem sido mais amigável e fácil de mensurar, pois seu modelo matemático possibilita a construção de árvores de decisões. Nestes modelos, a predição é formulada num espaço de busca com possíveis soluções (Faceli et al., 2011). Dentro desse conceito, várias abordagens já foram destacadas com o uso destes algoritmos e muitas vezes apresentam melhores desempenhos em termo de acurácia (Zalewski, 2015). As regras destes algoritmos são baseadas em uma proposta com a capacidade de divisão do espaço de instâncias em pequenos espaços e cada espaço dividido sofre alterações na base de um modelo (Faceli et al., 2011).

Para ilustração destes modelos, a figura 2.9 representa uma árvore de decisão para classificação, onde os nós representam os atributos com maiores ganhos de informações, e no final são apresentados os nós folhas que representam as classes previstas. Conforme citado

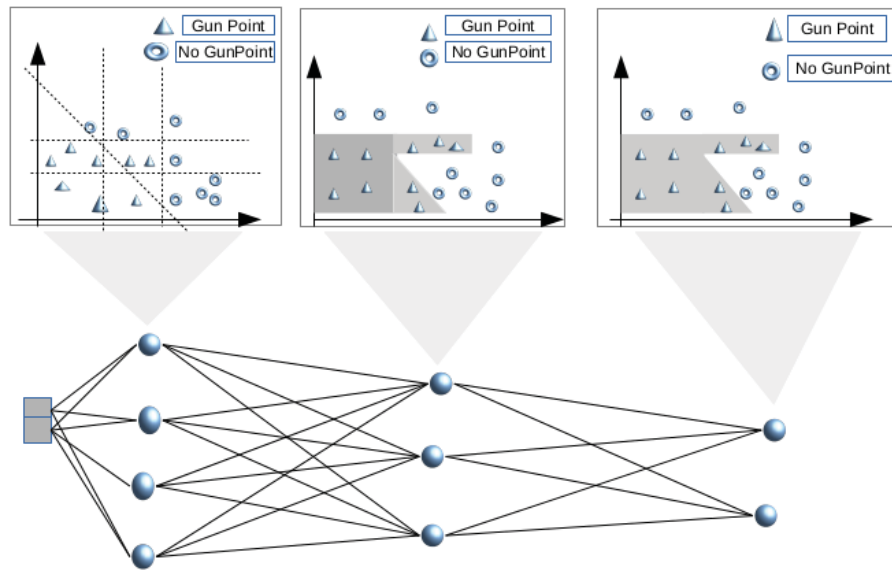


Figura 2.8: Impacto dos neurônios mediante seus desempenhos em diferentes camadas escondidas da rede MLP (Multilayer Perceptron), com base no conjunto de dados temporais (Gun/Point), uma adaptação de [Faceli et al. \(2011\)](#).

anteriormente, o objetivo não é criar modelos robustos para inferências pois os dados já foram pré-processados pelos modelos baseados em filtros.

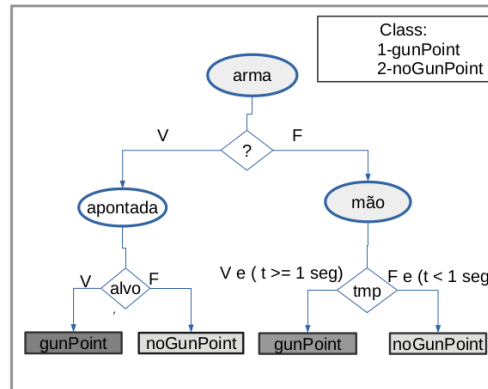


Figura 2.9: Ilustração do modelo de procura por meio de um problema de duas classes em forma de árvore de decisão.

2.4 Considerações Finais

Neste capítulo foram destacadas as tarefas de mineração de dados em séries temporais, com ênfase nos métodos de pré-processamento, intensificando o estudo das principais componentes de séries temporais e sua decomposição. Ainda nesta seção foi destacada a construção dos modelos preditivos adotados nesta pesquisa, bem como alguns exemplos de impacto mediante seus modos de inferências. Neste cenário, os modelos preditivos apresentam vantagens e desvantagens, de acordo com os problemas dos quais são induzidos. Portanto no próximo capítulo será demonstrado como ocorre a classificação de séries por meio da abordagem de *shapelet*.

Capítulo 3

Classificação Por Meio das *Shapelets*

3.1 Considerações Iniciais

Uma *shapelet* consiste em uma subsequência de série temporal que permite maximizar a discriminação entre as classes de um certo conjunto de dados. Nesse caso, para a construção dos modelos preditivos por meio de seu uso, é necessário quantificar a similaridade entre uma determinada *shapelet* e cada série temporal, possibilitando, assim, que a similaridade existente e detectada seja usada como característica discriminatória da série temporal (Lines et al., 2012).

Nesta abordagem, o uso das *shapelets* apresentam os seguintes benefícios na classificação de séries temporais:

1. Nos trabalhos de Pyle (1999) é destacada a compreensão e facilidade que a *shapelet* apresenta sobre o domínio de série temporal a ser mensurada.
2. Outro benefício que podemos destacar é a identificação de subsequências morfológicas similares que são independentes em relação a posição temporal (Lines et al., 2012) e (Hills et al., 2013).
3. O processo de classificação de novas instâncias acaba se tornando mais rápido devido a maneira compacta com que os classificadores são construídos (Keogh e Rakthanmanon, 2013).

Formalmente uma *Shapelet* é definida da seguinte maneira:

Definição 3.1 (Shapelet). Uma *Shapelet* S_H é alguma subsequência $S \in S_C$, que por meio de alguma métrica de qualidade permite uma melhor distribuição das séries temporais de um conjunto B^M para um dado conjunto C de classes.

Na figura 3.1 é demonstrada a apresentação de quatro séries temporais $\{T_1, T_2, T_3, T_4\}$, que se encontram distribuídas em duas classes (A e B), sendo destacada a *shapelet* que possibilita diferenciar as séries da classe A.

Inicialmente, o conceito de *shapelet* na sua essência foi pesquisado e proposto na literatura por Geurts (2001), o autor efetuou busca por subsequências que conseguissem discriminar uma certa classe. Nesta vertente, levou-se em consideração a busca das subsequências discriminantes como intratável, devido ao fator de tempo de processamento necessário.

No trabalho de Ye e Keogh (2009) é proposto um algoritmo como resultado de duas abordagens; a primeira baseia-se na redução da dimensão das séries, com propósito de encontrar uma determinada *shapelet* discriminante entre as demais: a segunda abordagem é proposta por

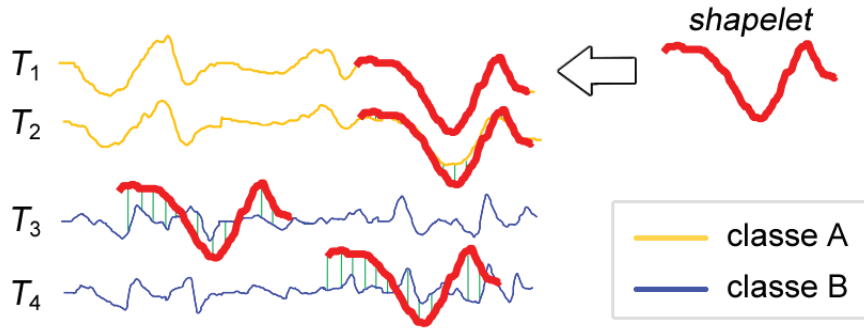


Figura 3.1: Ilustração da primitiva *shapelet* para a classificação de séries temporais (Ye e Keogh, 2009).

Yamada et al. (2003) com a finalidade de construir um modelo interpretável que pudesse realizar a extração de subsequências (Geurts, 2001), neste cenário o autor utilizara uma série temporal inteira para representar cada nó da árvore de decisão usado no modelo preditivo. Portanto, na discussão de Ye e Keogh (2009), o uso de características globais reduz a inteligibilidade do modelo preditivo. Neste contexto, foi proposto o algoritmo YK, este algoritmo emprega somente as subsequências mais discriminantes da série temporal para cada classe.

O Processo de identificação de *shapelet* de acordo com o algoritmo YK, é descrito no algoritmo 1, neste algoritmo são consideradas três etapas principais:

1. Gerar as subsequências;
2. Calcular a distância;
3. Determinar a qualidade.

Algoritmo 1 SelecionarMelhorShapelet (B^M, min, max)

Entrada: B^M : conjunto de M séries temporais, min : tamanho mínimo de subsequência, max : tamanho máximo de subsequência

Saída: *shapelet*: subsequência de melhor qualidade identificada

```

1 shapelet  $\leftarrow \emptyset$ 
  melhor  $\leftarrow 0$ 
   $S_C \leftarrow GerarSubsequenciasCandidatas(B^M, min, max)$ 
  para  $n \leftarrow min$  até  $max$  faça
2    para cada subsequência  $S$  em  $S_C$  faça
3       $D_S \leftarrow CalcularDistancias(S, B^M)$  qualidade  $\leftarrow DeterminarQualidade(D_S)$ 
4      se qualidade  $>$  melhor então
5        melhor  $\leftarrow qualidade$  shapelet  $\leftarrow S$ 
6      fim
  fim
7 fim
8 retorna shapelet

```

3.1.1 Gerar Subsequências Candidatas

O fundamento teórico para gerar todas as subsequências de um tamanho n é definido através de dois parâmetros no intervalo de [mínimo e máximo]. Estes parâmetros caracterizam o menor e o maior tamanho de possíveis subsequências respectivamente (Zalewski et al., 2016). Para gerar as subsequências possíveis é aplicado o conceito de janela deslizante formalmente descrito na definição 3.2.

Definição 3.2 (Janela Deslizante). *Seja uma série temporal T de tamanho m e seja n o tamanho das subsequências possíveis, uma janela deslizante de tamanho n consiste em um conjunto formado por todas as subsequências distintas de tamanho n que podem ser extraídas de T .*

O processo de construção de uma janela deslizante baseia-se em selecionar uma subsequência de tamanho n a partir de cada posição p da série temporal T , denotada por $S^{n,p} = \{t_p, \dots, t_{n+p-1}\}$, tal que $1 \leq p \leq (m - n + 1)$. Como resultado, o conjunto de $(m - n) + 1$ subsequências definidas pela janela deslizante pode ser expresso por:

$$S_T^n = \{S^{n,1} \cup S^{n,2} \cup \dots \cup S^{n,(m-n+1)}\} \quad (3.1)$$

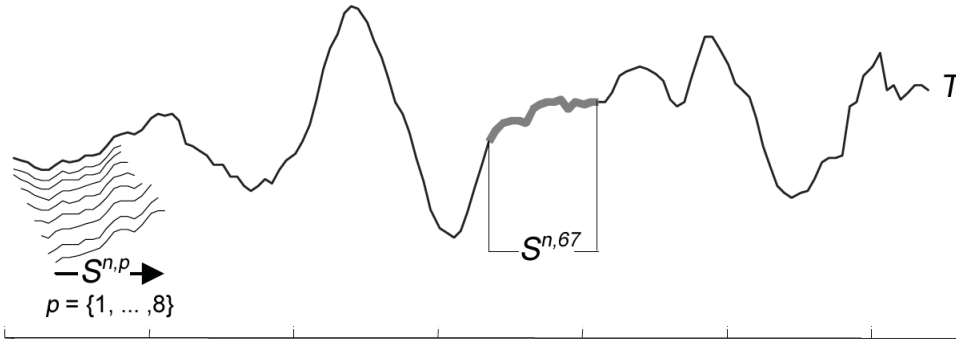


Figura 3.2: Ilustração do processo de aplicação de uma janela deslizante Keogh et al. (2003).

A quantidade de subsequências pertencentes a S^n (definido na equação 3.4) é dada por:

$$|S^n| = \sum_{T_i \in B^M} (m_i - n + 1) \quad (3.2)$$

m_i representa o tamanho da série temporal T_i .

Assumimos que, para um certo conjunto de dados B^M o tamanho n seja exclusivo nos intervalos $min \leq n \leq max$, verifica-se o conjunto de todas as subsequências candidatas da seguinte maneira:

$$S_C = \{S^{min} \cup S^{min+1} \cup \dots \cup S^{max}\} \quad (3.3)$$

Seja um conjunto de dados de treinamento B^M com M séries temporais, verifica-se o conjunto de todas as subsequências que são candidatas de tamanho n para uma série temporal $T_i \in B^M$ como S_i^n , tal que $1 \leq i \leq M$. Nesse sentido, verifica-se o conjunto de todas as subsequências possíveis de tamanho n em B^M como:

$$S^n = \{S_1^n \cup S_2^n \cup \dots \cup S_M^n\} \quad (3.4)$$

Quanto ao número total de elemento S_C dado por:

$$|S_C| = \sum_{n=\min}^{\max} \sum_{T_i \in B^M} (m_i - n + 1) \quad (3.5)$$

Algoritmo 2 GerarSubsequênciasCandidatas (B^M , min, max)

Entrada: B^M : conjunto de séries M temporais, \min : tamanho mínimo de subsequência, \max : tamanho máximo de subsequência

Saída: S_C : conjunto de subsequências candidatas

```

1  $S_C \leftarrow \emptyset$  salto  $\leftarrow 1$   $n \leftarrow \max$  enquanto  $n \geq \min$  faça
2   para cada série temporal  $T$  em  $B^M$  faça
3      $S_C \leftarrow S_C \cup S_T^n$ 
4    $n \leftarrow n - \text{salto}$ 
5 retorna  $S_C$ 
```

Na figura 3.2 é ilustrada a tarefa de aplicação de uma janela deslizando de tamanho n sobre uma dada série temporal T . Na porção inicial da figura são mostradas oito subsequências de tamanho n extraídas das posições $p = \{1, \dots, 8\}$; enquanto que na porção central é representada a subsequência de tamanho n extraída a partir da posição $p = 67$.

O processo de geração das subsequências candidatas é descrito no algoritmo 2. Neste método a variável *salto* é utilizada para determinar a variação do tamanho de subsequências a cada iteração do algoritmo, ou seja, é necessário estimar um valor para a variável em questão. No trabalho de Ye e Keogh (2009, 2011) foi adotado $\text{salto} = 1$.

Para os parâmetros \min e \max os autores do algoritmo YK recomendam utilizar os seguintes: $\min = 3$, definindo o tamanho da subsequência significativa, e $\max = m$ utilizando o tamanho completo da série.

3.1.2 Calcular a Distância

Dada uma subsequência $S_i^n \in S_C$, para cada série temporal $T \in B^M$ é calculado a distância como descreve o algoritmo 3.

Algoritmo 3 CalcularDistâncias (S , B^M)

Entrada: B^M : conjunto de séries M temporais, S : subsequência de tamanho n

Saída: D_S : conjunto de distâncias de S em relação a B^M

```

1  $D_S \leftarrow \emptyset$  para cada série temporal  $T$  em  $B^M$  faça
2    $D_S \leftarrow D_S \cup \text{SubDist}(S, T)$ 
3 fim
4 retorna  $D_S$ 
```

Definição 3.3 (Distância entre séries temporais). Sejam duas séries temporais T_a e T_b , de igual tamanho m , a função $\text{Dist}(T_a, T_b)$ retorna um valor real não negativo d , que expressa a distância entre as duas séries. (Zalewski, 2015).

No algoritmo YK, a função *SubDist* também pode ser utilizada para calcular a distância existente entre duas subsequências de mesmo tamanho. Neste contexto, a tarefa de identificação de *shapelet*, requer o cálculo de similaridade entre subsequências diferentes. Nesta vertente, a distância da subsequência é definida por:

Definição 3.4 (Distância de subsequência). *Sejam duas subsequências S e R de tamanho n_S e n_R , respectivamente, tal que $n_S < n_R$, e seja $S_R^{n_S}$ a janela deslizante de tamanho n_S aplicada sobre a subsequência R , a distância de subsequência é dada pela função $SubDist(S, R)$:*

$$SubDist(S, R) = \min(Dist(S, r)), r \in S_R^{n_S} \quad (3.6)$$

onde \min é uma função que retorna a menor distância entre S e todas as subsequências contidas em $S_R^{n_S}$. (Zalewski, 2015).

Na figura 3.3 é demonstrada a representação da posição da série onde ocorre a menor distância entre as subsequências S e R .

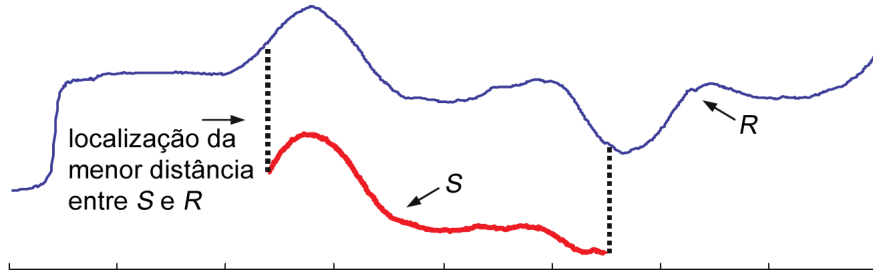


Figura 3.3: Representação esquemática da localização da menor distância encontrada entre as subsequências S e R Ye e Keogh (2011).

Finalmente o produto desta etapa é dado pela (definição 3.5) caracterizando uma certa subsequência em relação ao conjunto de séries temporais.

Definição 3.5 (Conjunto de distâncias). *Seja uma subsequência S de tamanho n e uma base de séries temporais B^M , composta por séries temporais de tamanho m , tal que $n \leq m$, o conjunto de distâncias de S é expresso por:*

$$D_S = \{d_1, d_2, \dots, d_M\} = \{SubDist(S, T_1), SubDist(S, T_2), \dots, SubDist(S, T_M)\} \quad (3.7)$$

3.1.3 Determinar a Qualidade

A terceira parte do algoritmo YK consiste em determinar a qualidade de uma subsequência S de um certo conjunto de distância D_S que foi calculado na etapa anterior. Neste caso, o conjunto de distância é transformado em uma linha de ordenação.

Algoritmo 4 DeterminarQualidade(D_S)

Entrada: D_S : conjunto de distâncias de S em relação a B^M
Saída: $maxGI$: máximo valor de ganho de informação

```

1  $L \leftarrow Ordenar(D_S)$   $maxGI \leftarrow 0$   $maxSep \leftarrow 0$ 
  para  $i \leftarrow 1$  até  $|L| - 1$  faça
2    $ls_i \leftarrow \frac{1}{2}(d_{i+1} + d_i)$   $GI \leftarrow Ganho(L, ls_i)$   $Sep \leftarrow DistSep(L, ls_i)$ 
     se  $(GI > maxGI)$  ou  $((GI = maxGI) \text{ e } (Sep > maxSep))$  então
3      $maxGI \leftarrow GI$   $maxSep \leftarrow Sep$ 
4   fim
5 fim
6 retorna  $maxGI$ 
  
```

A figura 3.4 apresenta um esquema de construção de uma linha de ordenação mencionada anteriormente, referente a uma subsequência.

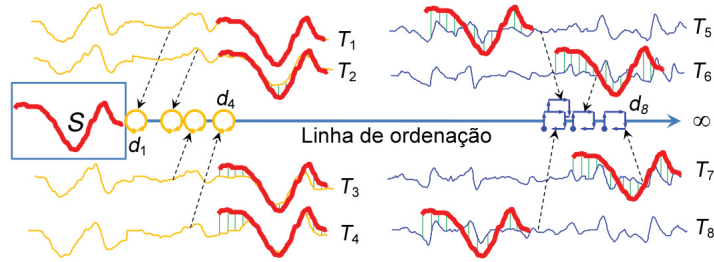


Figura 3.4: Representação esquemática de uma linha de ordenação Mueen et al. (2011).

Neste algoritmo, linha 5 do algoritmo 4, cada par adjacente dos elementos contidos na linha de ordenação L construída anteriormente é usada para determinar o valor do limiar de separação, de acordo com a definição 3.6.

Definição 3.6 (Limiar de Separação). Um limiar de separação ls é um número real que permite dividir uma linha de ordenação L em dois subconjuntos disjuntos $L_E = \{d_i : d_i \in L, d_i \leq ls\}$ e $L_D = \{d_i : d_i \in L, d_i > ls\}$.

Nesse processo a linha de ordenação L é dada por um conjunto de limiares de separação possíveis, representados do seguinte modo:

$LS = \{ls_1, ls_2, \dots, ls_{M-1}\}$, tal que:

$$ls_i = \frac{d_{i+1} + d_i}{2} \quad (3.8)$$

ilustramos através da figura 3.5 uma linha de ordenação constituída por nove séries temporais, pertencentes a duas classes, sendo cinco da classe círculo e quatro da classe quadrado. As posições assinaladas por ls_4 e ls_5 equivalem ao limiar de separação entre as séries temporais L_4 e L_5 , e L_5 e L_6 nessa ordem.

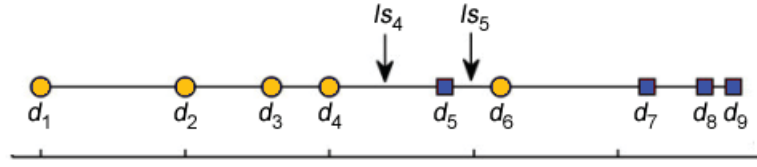


Figura 3.5: Ilustração do limiar de separação [Ghalwash e Obradovic \(2012\)](#).

Definição 3.7 (Ganho de Informação). Considerando um determinado limiar de separação ls , e seja o tamanho de L , L_E e L_D , expresso por $|L|$, $|L_E|$ e $|L_D|$ respectivamente; o ganho de informação é dado por:

$$\text{Ganho}(L, ls) = \text{Entropia}(L) - \left(\frac{|L_E|}{|L|} \text{Entropia}(L_E) + \frac{|L_D|}{|L|} \text{Entropia}(L_D) \right) \quad (3.9)$$

Definição 3.8 (Entropia). Seja uma linha de ordenação L , e seja n_k a quantidade de elementos de L que estejam associados a uma determinada classe $c_k \in C$, a entropia de L é expressa por:

$$\text{Entropia}(L) = \sum_{k=1}^w -\frac{n_k}{N} \log \frac{n_k}{N} \quad (3.10)$$

no qual $N = \sum_{j=1}^w n_j$.

Dadas as definições apresentadas acima, os autores de [Ye e Keogh \(2009\)](#) definem uma *shapelet* do seguinte modo:

Definição 3.9 (Shapelet). Uma *shapelet* é uma tupla $S_H = (S, ls)$, na qual S é alguma subsequência de S_C , que associada a um valor ls permite dividir uma linha de ordenação L em dois subconjuntos disjuntos, tal que o ganho de informação gerado por ls seja o máximo possível para L .

3.2 Considerações Finais

Neste capítulo foi apresentado como ocorre a classificação de séries temporais, através da primitiva *shapelet*. Foram demonstrados os principais fundamentos da primitiva *shapelet* e os algoritmos de identificação das *shapelet*, levando em consideração a qualidade central de cada *shapelet*, neste cenário as *shapelet* identificadas formam atributo-valor ([Lines et al., 2012](#)). No próximo capítulo apresentaremos os conceitos da transformada *shapelet* (TS) e uma ampla exploração mediante a utilidade que os algoritmos de mineração de dados oferecem, nestas avaliações serão destacados os fundamentos principais da TS, bem como os benefícios que eles apresentam para solucionar os problemas de classificação de séries temporais.

Capítulo 4

Explorando a Transformada *Shapelet*

4.1 Considerações Iniciais

Neste capítulo são detalhados os fundamentos da transformada *shapelet*, em que a *TS* resulta da execução dos algoritmos primitivos da identificação da *shapelet*. Como dito anteriormente, o uso da transformada *shapelet* possibilita a construção de diversos classificadores que a literatura apresenta para inferências de qualquer tipo de dados.

No estudo da transformada *shapelet* nota-se o mecanismo que possibilita possíveis soluções a respeito dos problemas definidos de classificação de séries temporais, pois as *shapelet* são utilizadas para a extração de características morfológicas de séries temporais (Zalewski, 2015). Portanto, apresentamos as extensões da transformada como ponto principal a ser explorado bem como nossas contribuições no estudos de CST, com o intuito de obter resultados promissores em variados problemas de séries temporais definidos.

4.2 Transformada *Shapelet*

Em Lines et al. (2012) os autores demonstraram a importância da separação dos algoritmos de classificação em relação a identificação das *shapelets*. Neste contexto, cada exemplo do conjunto de dados é constituído em uma nova representação distante aos dados originais. Na transformação desses dados são consideradas três etapas que podem ser realizadas e que formam os fundamentos da transformada *shapelet* segundo Lines et al. (2012); Hills et al. (2014) e Zalewski (2015):

Estimar o Parâmetro k

Esta fase, consiste em estimar de modo automático a quantidade de k *shapelets* que serão utilizadas, lembrando que k é o valor do limite máximo das *shapelets* peculiares que são extraídas. O parâmetro k não tem influência sobre a qualidade formal de cada *shapelet*, pois, é usado apenas como um limiar de corte. A estimativa do parâmetro k é fundamentada em Lines et al. (2012), onde foram proposto duas abordagens para estimar o valor de k . A primeira o valor de k é obtido por meio de $m/2$, sendo m o tamanho das séries. A segunda baseia-se por meio do processo de validação cruzada mediante um numero de partições sobre o conjunto de dados de treinamento. A partição de m que possibilitou maior média de acurácia é definido como o valor final de k .

Selecionar k Shapelet

Nessa fase é descrita pelo algoritmo 1, o qual apresenta o processo de seleção das *shapelets*. Em Ye e Keogh (2009) e Lines et al. (2012) no que tange ao processo de identificação das *shapelets*, a abordagem apresenta características distintas nos seguintes aspectos: a generalização da medida de qualidade e o processo de como as *shapelets* são manuseadas durante o processo de buscas.

Aplicar a Transformada

Na fase 3, dado um conjunto de k *shapelets* gerado na fase anterior, é utilizado com o para transformar um conjunto de dados temporais B^M e classes ($W_{classes}$) em uma representação atributo-valor. Nesse contexto, segundo os autores iniciais (Lines et al., 2012), as características discriminatórias em séries temporais não se encontram no domínio de tempo. Neste contexto, o valor do atributo é a distância entre a *shapelet* e a série original como é descrito na tabela 4.1.

Séries Temporais (T)	S_H^1	S_H^2	...	S_H^k	Classe (c)
T_1	d_{11}	d_{12}	...	d_{1k}	c_1
T_2	d_{21}	d_{22}	...	d_{2k}	c_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
T_M	d_{M1}	d_{M2}	...	d_{Mk}	c_w

Tabela 4.1: Representação atributo-valor da transformada *shapelet*.

A vantagem da transformação dos dados é pelo fato deste processo permitir que as *shapelets* sejam utilizadas por vários algoritmos de classificação (Hills et al., 2014). A desvantagem do algoritmo proposto por Lines et al. (2012) sobre a transformada *Shapelet* é notada ao estimar os parâmetros de tamanho [mínimo e máximo] a serem avaliados e o valor da variável k *Shapelet*, com o propósito de selecionar as melhores *Shapelet*. Em nossa pesquisa evitamos a estimativa desses parâmetros, uma vez que em vários casos a determinação dos parâmetros têm influência significativa sobre a qualidade dos modelos preditivos construídos a partir do algoritmo 1. Desse modo, focamos nossa pesquisa em uma das abordagens proposta por Zalewski et al. (2016), caracterizada como extensões da transformada *Shapelet*, denominada de *abordagem reduzida*.

4.2.1 Abordagem Reduzida TS+

Em Lines et al. (2012) e Hills et al. (2014) é realizada uma discussão interessante relativa à seleção de poucas *Shapelets* na transformada. Os autores concluem que, no uso desta metodologia, ao selecionar poucas *Shapelets* pode ser constatada a falta de informação para construção de modelos preditivos que possibilitam representar um certo domínio de dados temporais, ou sobre-ajuste (*Underfitting*). Portanto, é proposto no trabalho de Zalewski et al. (2016) um processamento posterior dedicado a filtragem das *Shapelets* identificadas, isto resulta em um novo subconjunto menor obtido mediante algum critério de qualidade, e este processamento é realizado por meio de algoritmos de seleção e agregação de atributos.

4.3 Abordagens da Transformada

Em [Zalewski et al. \(2016\)](#), os autores apresentam três abordagens inerentes a transformada com o intuito de melhorar o processo de seleção das melhores *shapelets* compostas de características discriminatórias das séries.

- **Abordagem Exaustiva - TSK:**

Esta abordagem de acordo a proposta de [Lines et al. \(2012\)](#), consiste na aplicação de um algoritmo que de modo exaustivo intensifica a busca de *shapelet* explorando os parâmetros mínimos e máximos do algoritmo original da primitiva. Essa abordagem tem por objetivo facilitar o processo de buscas considerando todos os tamanhos possíveis de subsequências ([Zalewski et al., 2016](#)).

- **Abordagem relaxada - TS***

A abordagem dos autores tem por estratégia utilizar as k melhores *shapelets*, isso é determinado por meio de duas técnicas, a primeira fixa com $k = m/2$, onde m corresponde ao tamanho da série e a segunda corresponde ao processo de validação cruzada aplicada no conjunto de treinamento, que tecnicamente automatiza o processo de estimativas do parâmetro k ([Zalewski et al., 2016](#)).

- **Abordagem reduzida - TS+**

Dentre as abordagens acima mencionadas, as pesquisas desenvolvidas no presente trabalho foram norteadas utilizando a abordagem reduzida TS+, que será discutida na próxima seção.

4.4 Criação das Reduzidas TS

Abordagem Reduzida TS+

A discussão anterior ([Lines et al., 2012](#); [Bagnall, 2012](#)) e [Zalewski et al. \(2016\)](#) a respeito da quantidade de *shapelets* selecionadas na transformada é resolvida nesta abordagem. O método empregado permite apenas a seleção de um subconjunto menor e uma pequena porção de *shapelet* que formam atributo-valor em cada conjunto de dados. O modelo tem os seguintes objetivos:

1. Permitir estimar de modo automático o parâmetro k .
2. Manter o processo de identificação das melhores *shapelets* separado em relação ao algoritmo original de classificação proposto por [Lines et al. \(2012\)](#)

Neste contexto são propostas três extensões da transformada *shapelet TS* em combinações com os algoritmos de seleção de atributos mencionados anteriormente, formados da seguinte maneira:

- Extensão (TS+CFS),
- Extensão (TS+CS),
- Extensão (TS+PCA), onde a TS representa a transformada *shapelet* mais o modelo de seleção de atributo respectivamente.

Através dessa proposta efetuamos as avaliações experimentais que serão discutidas no capítulo seguinte.

A redução da dimensionalidade em séries temporais pode ser tratada em aprendizagem de máquinas, através de algoritmos de seleção e agregação de atributos, ou por outras medidas de buscas de atributos com maior ganho de informação (Faceli et al., 2011). Neste contexto, filtrar atributos irrelevantes ou redundantes mediante algum critério de importância, possibilita que o processo ocorra no algoritmo de seleção adotado, buscando atributos com características discriminatórias na série temporal.

Portanto, na avaliação dos experimentos feitos neste estudo, utilizamos os seguintes algoritmos baseados nas abordagens de seleção e agregação de atributos: *Consistency-based Filter*(CS) Liu et al. (1996), *Correlation-based Feature Selection* (CFS), e um método de agregação *Principal Component Analysis* (PCA) Smith et al. (2002). Esses métodos de seleção de subconjunto de atributos são frequentemente usados e citados na literatura.

Por meio dos modelos de seleção de atributos mencionados anteriormente, com base nos resultados obtidos pelos algoritmos, propomos três extensões da TS a serem avaliadas, (*TS+CFS*, *TS+CS*, *TS+PCA*) com o propósito de selecionar novamente as melhores *shapelets*. Como dito anteriormente, nessa abordagem de combinação foram realizadas para todas as três variações busca heurística usando o algoritmo guloso *Best First* e a direção de busca *Forward*.

4.5 Configuração e Descrição dos Dados

A tabela 4.2 descreve a configuração dos conjuntos de dados para avaliação dos métodos propostos nesta pesquisa. Avaliamos 25 conjuntos de dados disponíveis no repositório *UCR* (Chen et al., 2015) estruturados em dados de treinamentos e testes, lembrando que muitos destes dados já foram definidos por vários pesquisadores que ofereceram os mesmos ao repositório anteriormente citado. Na mesma tabela é descrita a estrutura de cada conjunto de dados, onde W_{Classe} corresponde a quantidade de classes, M_{Train} é a quantidade de instâncias existentes na base de treinamento e M_{Test} é a quantidade de instâncias na base de testes, finalmente o *Tamanho* é o número de atributos que corresponde ao tamanho da série temporal.

4.6 Método Holdout

O método *holdout* particiona o conjunto de dados em dois subconjuntos exclusivos de B^M para treinamento e teste de acordo com a ilustração da figura 4.1. A desvantagem deste método é que o modelo preditivo é treinado apenas com os dados de treinamento, neste cenário o modelo pode fornecer certas tendências otimistas. No entanto, para (Faceli et al., 2011), em conjuntos de dados, a divisão não se torna um problema quando se trata de dados em grande escalas, e a comunidade de séries temporais alega que esta configuração dos dados é a mais adequada para solução dos problemas em séries temporais.

4.6.1 Configuração dos Experimentos

No cumprimento dos objetivos específicos desta pesquisa, os experimentos realizados referente à exploração da transformada *shapelet* estão constituídos em cinco fases conforme a ilustração da figura 4.2.

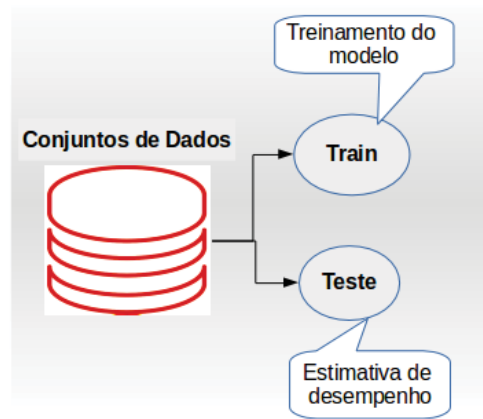


Figura 4.1: Configuração de amostragens dos dados, uma adaptação de [Faceli et al. \(2011\)](#).

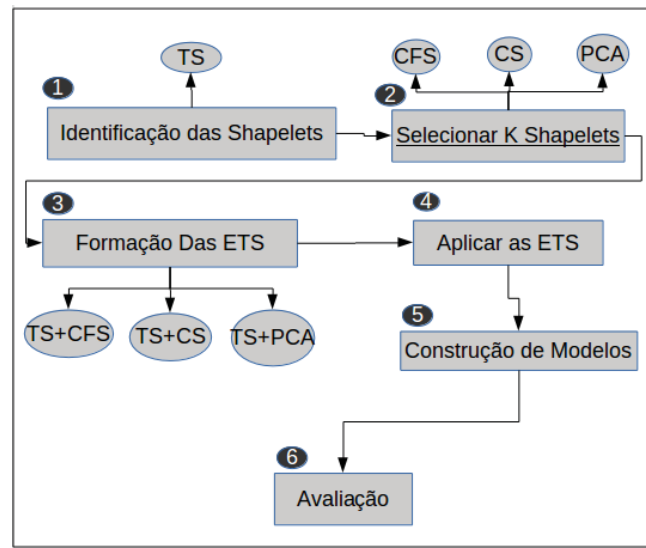
Tabela 4.2: Sumarização dos conjuntos de dados temporais.

#	Conjunto de dados	w	M_{treino}	M_{teste}	m	Tipo Série
1	<i>Adiac</i>	37	390	391	176	Imagem
2	<i>Beef</i>	5	30	30	470	Spectro
3	<i>CBF</i>	3	30	900	128	Simulação
4	<i>ChlorineConcentration</i>	3	467	3840	166	Simulação
5	<i>Coffee</i>	2	28	28	286	Espectro
6	<i>DiatomSizeReduction</i>	4	16	306	345	Imagem
7	<i>ECG200</i>	2	100	100	96	ECCG
8	<i>ECGFiveDays</i>	2	23	861	136	Dispositivo
9	<i>FaceAll</i>	14	560	1690	131	Imagem
10	<i>FaceFour</i>	4	24	88	350	Imagem
11	<i>FacesUCR</i>	14	200	2050	131	Espectro
12	<i>Fish</i>	7	175	175	463	Imagem
13	<i>GunPoint</i>	2	50	150	150	Movimento
14	<i>ItalyPowerDemand</i>	2	67	1029	24	Dispositivo
15	<i>Lighting2</i>	2	60	61	637	Sensor
16	<i>Lighting7</i>	7	70	73	319	Sensor
17	<i>MedicalImages</i>	10	381	760	99	Imagem
18	<i>MoteStrain</i>	2	20	1252	84	Sensor
19	<i>OliveOil</i>	4	30	30	570	Imagem
20	<i>OSULeaf</i>	6	200	242	427	Imagem
21	<i>SonyAIBORobotSurface</i>	2	20	601	70	Sensor
22	<i>SwedishLeaf</i>	15	500	625	128	CCG
23	<i>SyntheticControl</i>	6	300	300	60	Simulação
24	<i>Trace</i>	4	100	100	275	Sensor
25	<i>TwoLeadECG</i>	4	1000	4000	128	CCG

(1) Identificação das shapelets

Esta fase realiza o processo de identificação das *shapelets* nos dados brutos temporais com o propósito de constituir a transformada *shapelet* ([Lines et al., 2012](#)).

Figura 4.2: Representação esquemática dos experimentos realizados de acordo com a exploração da transformada *shapelet*.



(2) Selecionar k shapelets

Esta etapa realiza a busca das melhores *shapelets* com características discriminatórias de acordo com os métodos propostos.

(3) Formação das ETS

De acordo com as *shapelets* selecionadas na etapa (2), para cada extensão da transformada, são constituídos conjuntos de treinamentos e de testes formando uma representação atributo-valor.

(4) Aplicar as ETS

Na fase 4, o conjunto de k *Shapelet* gerado no processo anterior é utilizado para transformar um conjunto de séries temporais B^M de w classes em uma representação atributo-valor, conforme a tabela 4.2.

(5) Construção dos Modelos

Nesta etapa foram construídos os modelos de predição por meio dos algoritmos propostos, os quais foram utilizados usando os parâmetros padrão do *framework Weka* (Hall et al., 2009). Lembrando que o propósito das extensões criadas é o de possibilitar a construção de classificadores, sem a necessidade de ajustes de parâmetros ou maximização de uma função no modelo preditivo.

(6) Avaliação

A avaliação foi realizada em duas fases: a primeira corresponde à avaliação dos modelos de seleção das *shapelets* em termo de quantidade, isto é, a busca pelos modelos que tiveram como resultado um número mais reduzido de *shapelet* em relação aos demais. Na segunda avaliação foi utilizada a acurácia como medida de desempenho para avaliação dos modelos preditivos, bem como a abordagem proposta para solução dos problemas de classificação de séries temporais com uma análise comparativa com o estado-da-arte.

4.7 Análises Estatísticas

Através das análises estatísticas é possível detectar as diferenças existentes entre o desempenho dos n modelos preditivos adotados. Nestas pesquisas sobre os M conjuntos de dados, foi utilizada como recomendação a aplicação do teste de *Friedman* (Demsar, 2006), sendo o teste uma versão não paramétrica correspondente a ANOVA medidas repetidas para analisar os dados que são pareados. Nesse contexto a comparação de posições é estabelecida em forma de *rank*. Entretanto, cada conjunto de dados e cada modelo são associados a uma posição com base no seu desempenho, numa ordenação dos melhores colocados para os piores. Caso ocorram empates serão atribuídos valores médios de posição.

O *rank* médio é calculado por meio da equação 4.1, assim, seja R_{ij} a posição de desempenho do modelo j , tal que $1 \leq j \leq K$ para um conjunto de dados i , tal que $1 \leq i \leq N$.

$$R_j = \frac{\sum_{i=1}^N R_{ij}}{N} \quad (4.1)$$

Quanto a hipótese nula de que os modelos que foram comparados são iguais, apresentando valores de *rank* iguais, a análise estatística de *Friedman* é representada pela equação 4.2.

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right] \quad (4.2)$$

Em Iman e Davenport (1980) é recomendada uma proposta de uma versão do teste de *Friedman* distribuída mediante a fórmula F com $(K-1)$ e $(K-1)(N-1)$ graus de liberdade. Esta proposta é expressa pela equação 4.3.

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)\chi_F^2} \quad (4.3)$$

Se a hipótese nula é rejeitada pelo teste de *Friedman*, isto denota a existência de diferenças significativas de desempenhos entre os modelos, mas o teste não permite descrever quais dos modelos realmente apresentam diferenças estatísticas. Neste cenário, o pós-teste de *Nemenyi* (Nemenyi, 1963) é recomendado com a finalidade de detectar diferenças significativas entre os modelos (Demsar, 2006). O cálculo para diferença crítica (CD) é dada pela equação 4.4.

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}} \quad (4.4)$$

Na tabela 4.3 são apresentados os valores de q_α com $\alpha = 0,05$ para diferentes quantidades de métodos a serem comparados, utilizando os pós-testes de *Nemenyi* e de *Bonferroni-Dunn*, de acordo com (Demsar, 2006).

Tabela 4.3: Valores de $q_{0,05}$ para diferentes valores de K no pós-teste de *Nemenyi* e de *Bonferroni-Dunn* (valores extraídos em Demsar (2006)).

K	2	3	4	5	6	7	8	9	10
Nemenyi	1,960	2,343	2,569	2,728	2,850	2,949	3,031	3,102	3,164
Bonferroni-Dunn	1,960	2,241	2,394	2,498	2,576	2,648	2,690	2,724	2,773

Na presença de vários modelos, são realizadas várias comparações em relação ao único modelo que se pretende destacar. Estatisticamente é recomendado o pós-teste de *Bonferroni-Dunn*

(Dunn e Dunn, 1961), onde q_α em algumas situações pode ser menos restrito. Neste cenário são feitas $K - 1$ comparações, resultando num maior nível de certeza e robustez no modelo estatístico.

4.8 Avaliação de Resultados das ETS

Os resultados obtidos para cada extensão da *TS* são considerados promissores, pois as *shapelets* foram reduzidas consideravelmente. Esta redução e a simplificação das séries temporais é constatada em cada variação como descreve o gráfico 4.3, o eixo dos x indicam os n conjuntos de dados e no eixo dos y o número de *shapelets* selecionadas por cada extensão. A tabela 4.4 descreve as extensões (TS+CFS, TS+CS e TS+PCA) e seus resultados em função da combinação com a transformada *shapelet* (*TS*) nos resultados obtidos nos conjuntos de treinamento.

Tabela 4.4: Resultados das melhores *shapelets* selecionada pelas extensões TS+. Os valores estão representados em formato de números inteiros, cada número indica uma *shapelet*. As células destacadas representam as quantidades menos reduzidas de *shapelets* em relação as demais extensões.

#	Extensões da TS		
	TS+CFS	TS+CS	TS+PCA
1	16	3	1
2	49	2	20
3	15	1	22
4	7	3	5
5	127	1	17
6	6	1	11
7	7	5	8
8	40	1	16
9	32	7	10
10	8	1	16
11	10	6	4
12	23	3	10
13	23	1	16
14	2	1	3
15	126	3	51
16	33	5	37
17	10	8	6
18	40	1	10
19	36	1	19
20	31	9	24
21	35	1	11
22	19	15	1
23	10	4	2
24	27	1	21
25	41	1	11

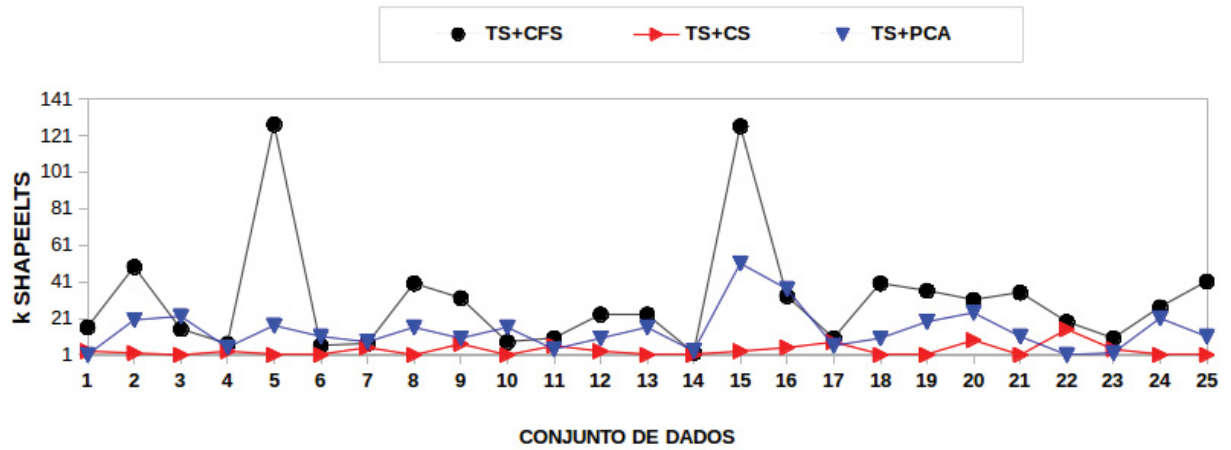


Figura 4.3: Variação quantitativa das melhores k shapelet selecionadas sobre os 25 conjuntos de dados mediante as extensões $TS+CFS$, $TS+CS$ e $TS+PCA$.

Mediante a análise de diferenças críticas, por meio da aplicação do teste de Friedman com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 4.4, a hipótese nula de que todos os modelos são equivalentes, em termos de acurácia, foi rejeitada com valores de $F_F = 37,47$ e $p\text{-valor} = 0,0001$. Desse modo, para verificar quais são as diferenças significativas, o pós-teste de Nemenyi foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0,66$.

A figura 4.4 ilustra o diagrama de diferença estatística existente entre as extensões da transformada *shapelet* (TS), enquanto que a figura 4.3 apresenta a variação das extensões em termo de quantidade de shapelet selecionadas em cada conjunto de dados.

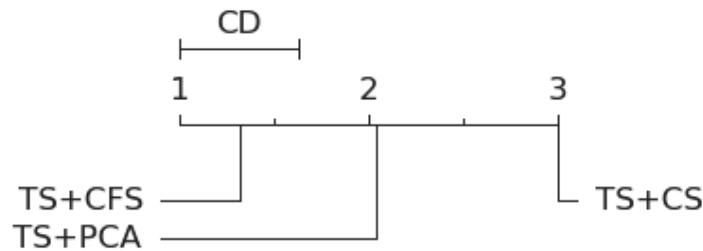


Figura 4.4: Diagrama de diferenças críticas através do *rank médio* obtido pelas extensões da TS em relação aos 25 conjuntos de dados.

As extensões propostas nesta pesquisa apresentam características similares, pois, o empenho pela redução brusca das séries e a busca de um número reduzido de *shapelet* é intensificada nos demais conjuntos de dados como é denotado nos resultados da tabela 4.4 em relação a tabela 4.2.

Nas discussões de Lines et al. (2012), Hills et al. (2014), e Zalewski et al. (2016) os autores apresentam a relevância do ganho ou perda de informações, ao selecionar poucas *shapelets* ou uma quantidade maior para formar novos subconjuntos de dados transformados, isso pode resultar na falta de informações ou sobreajustes de dados influenciando no modelo de classificação.

Diante disso, ressaltamos, que em nossos experimentos, a quantidade de *shapelet* selecionada que formam novos subconjuntos de dados é inferior em relação ao tamanho original

da série temporal. Quando o número de atributos é maior pode ocasionar um sobreajustes no modelo de classificação, enquanto que na utilização de poucas *shapelet* pode haver falta de informação para inferência. Conforme dito anteriormente, um dos nossos objetivos é a redução significativa do tamanho das séries sem que haja perda de informações para classificação, portanto, o impacto dos resultados obtidos como descreve a tabela 4.4 será discutido nas próximas seções.

4.9 Considerações Finais

Neste capítulo foram descritos os fundamentos da transformada *shapelet* por meio das maiores vantagens e desvantagens constatadas nesta abordagem, especificamente o método de busca das k melhores *shapelets* com características discriminatórias nas séries temporais. Mediante as dificuldades constatadas na *TS*, espera-se que os modelos proporcionados em MD mencionados anteriormente, para representação de séries temporais, sejam suficientes para exploração da *TS* com o propósito de simplificar a complexidade das séries temporais e selecionar apenas uma pequena porção de *shapelets* com características discriminatórias nas séries temporais. Através das extensões da *TS* (*TS+CFS*, *TS+CS* e *TS+PCA*) em termo de quantidade de *shapelet* foi constatada uma redução das séries e apenas um número reduzido de *shapelets* que formam atributo-valor. Ainda neste capítulo foram descritos os fundamentos das análises estatísticas utilizadas nesta pesquisa para detectar diferenças críticas entre os modelos em termo de desempenhos em uma avaliação experimental, por intermédio do pós-teste, em que foi constatado a existência de diferenças críticas entre estas extensões de acordo com a quantidade de *shapelet*. Portanto, no próximo capítulo será feita uma avaliação da transformada *shapelet* por meio dos modelos de predição construídos, com o propósito de demonstrar que a transformada *shapelet* também pode prover melhores resultados em termo de acurácia comparados com o resultados estado-da-arte.

Capítulo 5

Avaliação Experimental das *ETS*

5.1 Considerações Iniciais

Este capítulo descreve as avaliações das extensões da transformada *shapelet* onde serão destacados os desempenhos obtidos em cada extensão com base na diversidade de modelos preditivos experimentalmente adotados. Ainda nesta seção será apresentada uma análise de como os algoritmos foram configurados por meio da indução dos conjuntos de dados. Portanto, antes do processo de classificação, apresentaremos uma análise visual da dispersão de alguns conjuntos de dados, com intuito de demonstrar o quanto a distância entre os exemplos pertencentes a classes diferentes foi maximizada de acordo com as extensões da transformada *shapelet*.

5.2 Resultados de Agrupamentos

De modo aleatório apresentamos os resultados de agrupamento de quatro conjuntos de dados definidos anteriormente. Conforme mencionado, o agrupamento de séries temporais ou outro tipo de dados consiste na intensificação da similaridade existente entre eles, ou seja, dados similares tendem a estar mais próximos. Antes da indução dos dados nos modelos preditivos de acordo com a configuração de cada extensão, demonstramos de modo visual por intermédio do uso do algoritmo *K-means* a dispersão dos exemplos em cada extensão.

As figuras 5.1 e 5.2 ilustram os resultados de agrupamentos das extensões *TS+CFS*, *TS+CS* e *TS+PCA*, por meio de suas métricas de limpeza de ruídos e de dados redundantes, bem como a intensificação da redução de dados que possibilitam a facilidade de agrupamento e melhor visualização da dispersão. Ainda nessas figuras nota-se que cada extensão teve uma estrutura de representação dos dados transformados de modo diferente, consequentemente, isto implica no resultado final de agrupamento, pois em cada extensão os dados apresentam uma dispersão ou formação de grupos diferentes. Porém, todas as extensões sofreram o impacto da medida de proximidade entre pares de exemplos, esta proximidade é nada mais que a medida de similaridade ou dissimilaridade. Normalmente estas medidas consideram que todos os atributos são iguais e contribuem de igual modo para o cálculo da medida (Faceli et al., 2011) e isto está de acordo com as abordagens das extensões criadas, pois nesta nova representação dos dados, cada atributo corresponde a uma *shapelet* com características discriminatórias nas séries temporais.

Portanto, o impacto desta dispersão será constatado de acordo com os resultados dos modelos preditivos construídos na avaliação em termo da acurácia.

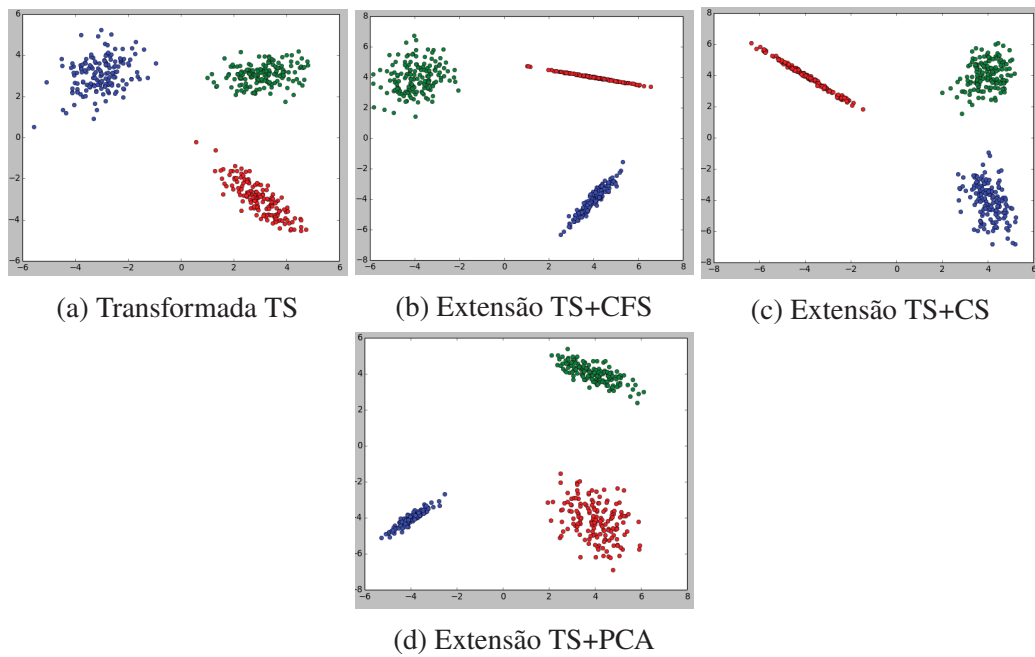


Figura 5.1: Resultados de agrupamento do conjunto *Chlorine Conntrations*.

● Classe A ● Classe B ● Classe C

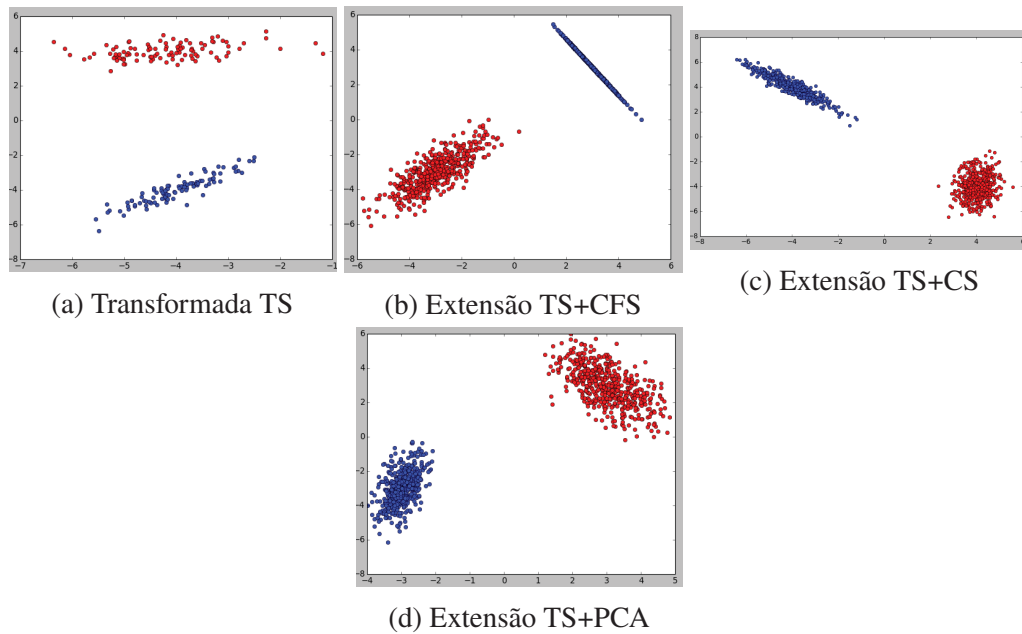


Figura 5.2: Resultados de agrupamento do conjunto de dados *Gun Point*.

● Classe A ● Classe B

5.3 Configuração dos Modelos Preditivos

Conforme já relatado, neste trabalho foram construídos modelos preditivos baseados em diversos métodos de inferências com o propósito de explorar a diversidade dos classificadores. Nesse contexto, a figura 5.3 descreve a configuração dos experimentos da seguinte maneira:

1. Tendo os conjuntos de dados da transformada shapelet (TS) numa divisão de treinamento e teste como ilustra a figura 4.1, esses dados são inseridos nos modelos de seleção de atributos (*CFS*, *CS* e *PCA*).
2. Cada modelo de seleção de atributo seleciona uma pequena porção de *shapelet* que forma novamente atributo-valor. Nesse cenário as extensões da transformada TS estão construídas da seguinte maneira: (*TS+CFS*, *TS+CS* e *TS+PCA*).
3. Os conjuntos de dados formados são induzidos aos modelos de predição de diversas famílias de classificadores em cada modelo (*Mod1*, *Mod2*, *Mod3*, *Mod4*) onde estas famílias podem ser constituídas por mais de um modelo preditivo para classificação das séries temporais. Cada modelo **Mod** é composto pelos seguintes algoritmos:
 - (a) *Mod1* é composto pelos algoritmos baseados em distância (IBK).
 - (b) *Mod2* é composto pelos algoritmos baseados em probabilidade (Naive bayes e rede bayesiana).
 - (c) *Mod3* é composto pelos modelos baseados em otimização (máquina de vetores de suportes (svm) e as redes neurais artificiais (MLP)).
 - (d) *Mod4* é composto pelos modelos baseados em procura (árvores de decisão J48 e o Random Forest).
4. Finalmente surge a predição (class) realizada pelos modelos . Cada modelo pode predizer de modo diferente a que classe uma determinada série temporal pertence.

Para avaliação dos modelos propostos foi utilizada a acurácia ou taxa de acerto como sendo a medida de avaliação de desempenho dos classificadores. Nas próximas seções são destacados os resultados parciais de cada extensão da transformada, considerando o número reduzido de *shapelet* selecionadas.

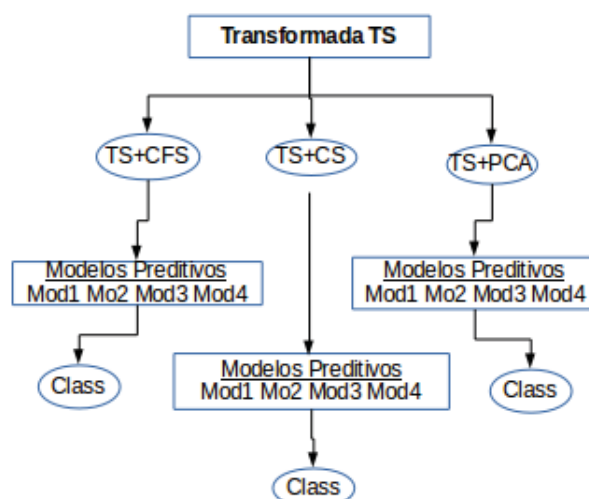


Figura 5.3: Configuração das extensões nos modelos preditivos.

5.4 Avaliação

Nessa avaliação dos resultados experimentais foi realizada uma análise de desempenho dos modelos de classificação levando em consideração a acurácia obtida. Ao explorarmos as extensões da *TS*, a predição de cada classe nas séries temporais foi baseada no método *holdout*, esse método consiste em particionar o conjunto de dados em duas partes, sendo um conjunto para treinamento e o outro para testes, como consta na sumarização da tabela 4.2.

De acordo com as matrizes de confusão geradas pelos classificadores, várias medidas de desempenhos dos modelos de predição podem ser construídas. Portanto, nessas matrizes destacam-se as seguintes variáveis:

1. *acc*: representa a acurácia.
2. *VP*: corresponde ao número de exemplos pertencentes a classe positiva classificados corretamente.
3. *VN*: iguala-se ao número de exemplos da classe negativa classificados corretamente.
4. *n*: corresponde ao número total de exemplos pertencentes nas duas ou mais classes, inerentes a um determinado conjunto de dados.

Portanto, deduz-se a seguinte equação para o cálculo da acurácia:

$$acc(f) = \frac{VP + VN}{n} \quad (5.1)$$

Para esta avaliação propusemos os seguintes classificadores; os baseados em distâncias são representados pelos algoritmos: *IBK*, *INN* *IB1*, enquanto que os modelos baseados em probabilidades são: *Naive Bayes* e *Bayes Net*. Para os modelos baseados em árvore de decisão utilizamos os seguintes algoritmos: *J48* e *Random Forest*, enquanto que, para os modelos baseados em otimização destacam-se os seguintes classificadores: *SVM* e *MLP*.

5.4.1 Avaliação da Extensão *TS+CFS*

Mediante o *pool* de classificadores construídos anteriormente, considera-se a acurácia como medida de avaliação dos algoritmos de predição. Nessa seção apresentamos os resultados de valores obtidos pela extensão *TS+CFS*. A tabela 5.1 descreve os valores em acurácia e um *rank* atribuído a cada algoritmo. Conforme a discussão de (Lines et al., 2012), (Bagnall, 2012) e (Zalewski, 2015) os algoritmos baseados em distâncias (1 vizinho mais próximo) são considerados estado-da-arte para CST. A extensão da transformada *shapelet TS+CFS* apresenta resultados equiparados aos modelos de diferentes predição, isto implica dizer que as *shapelet* selecionadas anteriormente representam fortemente as séries temporais que se pretende classificar.

Em relação aos *n* conjuntos de dados todos os algoritmos atingem mais de três vitórias, conforme o *rank* atribuído a eles. Ainda nesta análise, a extensão em estudo promove a construção de modelos preditivos inteligíveis para CST. Em termos de resultados o classificador *Random Forest* aparece em primeiro lugar com a acurácia média de 87,99%, em último está o classificador *Multilayer Perceptron* com 85,48%.

Com poucas *shapelets* selecionadas, em cada conjunto de dados, a extensão *TS+CFS* destaca-se como uma das metodologias adequadas para generalização de certo conjunto de dados temporais que se pretende transformar. Esta abordagem apresenta um padrão diferente de seleção das *shapelet* em relação as demais extensões, pois o modelo permite que a quantidade de *shapelet*

selecionadas que formam atributo-valor seja suficiente para análise e a predição por qualquer modelo de classificação inteligível.

Tabela 5.1: Resultados dos experimentos realizados para a avaliação da TS+CFS. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e o *rank* é a posição relativa do algoritmo em relação aos demais para um certo conjunto de dados. As células destacadas representam os melhores valores de acurácia obtidos para cada conjunto de dados.

#	IBK	NB	SVMl	MLP	Rand For
1	98,72 (1)	96,93 (3)	84,40 (4)	80,05 (5)	98,47 (2)
2	66,67 (3,5)	66,67 (3,5)	63,33 (5)	76,67 (2)	80,00 (1)
3	97,00 (4)	98,89 (1)	97,78 (3)	98,00 (2)	94,56 (5)
4	99,25 (4)	93,18 (5)	99,51 (3)	99,69 (2)	99,82 (1)
5	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	96,43 (5)	100,00 (2,5)
6	79,09 (2)	87,91 (1)	68,63 (5)	76,47 (3)	75,49 (4)
7	84,00 (1)	81,00 (4)	79,00 (5)	83,00 (2)	82,00 (3)
8	99,42 (3)	98,72 (4)	99,88 (1)	99,65 (2)	97,45 (5)
9	75,80 (3)	74,08 (4)	78,94 (1)	73,61 (5)	76,69 (2)
10	94,32 (2)	82,96 (4,5)	82,96 (4,5)	96,59 (1)	90,91 (3)
11	85,27 (4)	84,93 (5)	87,85 (1)	85,85 (2,5)	85,85 (2,5)
12	96,57 (4,5)	97,71 (1,5)	96,57 (4,5)	97,14 (3)	97,71 (1,5)
13	100,00 (2)	99,33 (4)	100,00 (2)	100,00 (2)	98,00 (5)
14	96,02 (1)	93,68 (3)	92,32 (4)	93,88 (2)	89,99 (5)
15	52,46 (5)	59,02 (2,5)	57,38 (4)	59,02 (2,5)	63,93 (1)
16	83,56 (2)	76,71 (4)	86,30 (1)	80,82 (3)	75,34 (5)
17	81,71 (4)	66,71 (5)	88,29 (3)	90,26 (1)	89,61 (2)
18	91,69 (3)	93,29 (1)	92,65 (2)	90,90 (4)	86,34 (5)
19	100,00 (2)	100,00 (2)	93,33 (5)	96,67 (4)	100,00 (2)
20	80,17 (5)	84,30 (1)	82,23 (4)	83,47 (2,5)	83,47 (2,5)
21	86,52 (4)	78,87 (5)	88,02 (3)	95,84 (1)	88,69 (2)
22	89,12 (2)	82,72 (4)	77,76 (5)	91,04 (1)	86,40 (3)
23	98,67 (2)	97,33 (4)	99,33 (1)	98,33 (3)	97,00 (5)
24	98,00 (3)	98,00 (3)	98,00 (3)	98,00 (3)	98,00 (3)
25	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	99,03 (5)
Media Acc	87,64	85,79	85,80	87,48	87,99
Rank Médio	2,88	3,20	3,16	2,64	3,12

Por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 5$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 5.1, não foi possível rejeitar a hipótese nula com valores de $F_F = 0,54$ e $p\text{-valor} = 0,7066$. Neste caso, não há diferença estatística entre os algoritmos nesta extensão. Essa análise evidencia a capacidade da extensão TS+CFS em realizar o processo de busca das melhores *shapelets* de maneira correta, permitindo que os modelos preditivos adotados possam atingir resultados significantes em termo de acurácia. Portanto, o modelo fornece maior facilidade para a classificação de séries temporais.

5.4.2 Avaliação da Extensão TS+CS

A extensão *TS+CS* apresenta uma redução brusca das *shapelets*, ou seja, o tamanho da série temporal é drasticamente reduzido. Outro fenômeno que se pode constatar é o fato dessa extensão não explorar ao máximo todas as subsequências possíveis das séries, a redução em causa influencia diretamente no processo de classificação, ressaltando que, o modelo em análise é mais eficiente nas inferências de dados em grande escala.

A tabela 5.2 apresenta os resultados obtidos por esta extensão em relação aos n conjuntos de dados mediante os algoritmos preditivos citados anteriormente. Os valores encontram-se em formato de acurácias e um *rank* atribuído ao algoritmo em relação aos demais com base nos dados usados.

A média de acurácia atingida pelos modelos preditivos é inferior em relação a abordagem anteriormente analisada (*TS+CFS*), em que apenas o algoritmo *Random Foreste* chega próximo da média atingida pela extensão anterior, isto decorre por ser um classificador robusto com mecanismo de inferência eficaz. Com uma média de 83,14% este algoritmo acaba sendo o melhor opção para essa abordagem, portanto, alguns resultados alcançados nesta extensão não são totalmente satisfatórios.

Em continuação com a avaliação desta abordagem (*TS+CS*), por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 5$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 5.1, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia, foi rejeitada com valores de $F_F = 10.19$ e $p\text{-valor} = 0,00001$. Desse modo, o resultado estatístico aponta diferença significativa existentes entre os modelos. Para verificar quais são as diferenças significativas nos algoritmos nessa extensão, o pós-teste de *Naminye* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 1.11$.

A diferença crítica (CD) é mostrada na parte superior do eixo das abscissas partindo da esquerda, como ilustrado na figura 5.4.

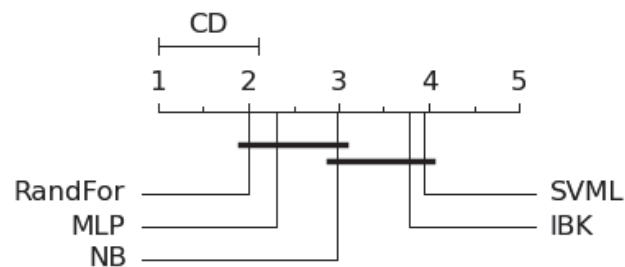


Figura 5.4: Diagrama de diferença crítica para o pós-teste de *Naminye* aplicado nos resultados da *TS+CS*.

5.4.3 Avaliação da Extensão TS+PCA

Na avaliação da extensão *TS+PCA* nota-se resultados bons de acurácia, deve-se à influência do método empregado para seleção das k melhores *shapelets*. Esse método não realiza somente a seleção da *shapelet*, o mesmo agrega valores de características principais a serem utilizadas para predição. Esta extensão é construída através de um modelo clássico usado em vários problemas, especificamente na solução da maldição da dimensionalidade de dados (Faceli et al., 2011).

Tabela 5.2: Resultados dos experimentos realizados para a avaliação 2 mediante a abordagem *TS+CS*. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um certo conjunto de dados. As células destacadas representam os melhores valores de acurácia obtidos em cada conjunto.

#	IBK	NB	SVMl	MLP	Rand For
1	92,07 (3)	95,40 (2)	26,59 (5)	46,04 (4)	97,44 (1)
2	33,33 (5)	60,00 (3)	53,33 (4)	63,33 (1,5)	63,33 (1,5)
3	45,56 (5)	85,22 (3)	64,00 (4)	92,00 (2)	93,00 (1)
4	95,16 (4)	94,38 (5)	98,54 (3)	98,85 (2)	99,01 (1)
5	92,86 (3,5)	92,86 (3,5)	82,14 (5)	96,43 (1,5)	96,43 (1,5)
6	48,37 (4)	52,61 (3)	30,72 (5)	61,77 (1)	61,11 (2)
7	80,00 (3,5)	81,00 (2)	80,00 (3,5)	79,00 (5)	82,00 (1)
8	67,25 (4)	97,45 (3)	50,41 (5)	98,03 (2)	98,49 (1)
9	70,24 (4,5)	70,24 (4,5)	73,43 (1)	71,36 (3)	72,96 (2)
10	45,46 (4)	84,09 (1)	44,32 (5)	70,46 (3)	80,68 (2)
11	76,78 (5)	81,46 (3)	79,02 (4)	81,56 (2)	82,83 (1)
12	81,71 (4)	89,14 (2)	80,00 (5)	84,57 (3)	90,29 (1)
13	87,33 (3)	84,67 (4)	78,00 (5)	92,67 (1,5)	92,67 (1,5)
14	94,95 (1)	89,70 (3)	51,51 (5)	90,28 (2)	89,12 (4)
15	57,38 (4)	60,66 (1,5)	59,02 (3)	60,66 (1,5)	54,10 (5)
16	67,12 (4)	68,49 (3)	73,97 (1)	63,01 (5)	71,23 (2)
17	82,24 (4)	63,55 (5)	87,11 (2)	85,92 (3)	88,68 (1)
18	82,11 (4)	85,22 (1)	59,82 (5)	85,14 (2,5)	85,14 (2,5)
19	36,67 (5)	100,00 (2)	56,67 (4)	100,00 (2)	100,00 (2)
20	74,38 (5)	80,99 (3)	81,41 (2)	81,82 (1)	76,03 (4)
21	61,90 (4)	85,19 (3)	47,26 (5)	87,02 (1)	86,02 (2)
22	89,92 (1)	82,88 (4)	73,60 (5)	88,96 (2)	87,36 (3)
23	25,03 (5)	83,04 (4)	96,67 (2)	97,00 (1)	95,67 (3)
24	96,00 (4)	98,00 (2)	42,00 (5)	98,00 (2)	98,00 (2)
25	97,45 (1)	92,89 (4)	88,76 (5)	96,49 (3)	96,75 (2)
Média Acc	71,63	79,47	66,33	78,58	83,14
Rank Médio	3,78	2,98	3,94	2,30	2,00

A tabela 5.3 descreve os resultados de acurácias obtidos pelos algoritmos já mencionados e utilizados nesta pesquisa. Também é atribuído um *rank* com base na posição do algoritmo em relação aos demais sobre os n conjuntos de dados em experimentos.

Voltando na discussão anterior de (Ye e Keogh, 2011), (Bagnall, 2012) (Hills et al., 2014) e (Zalewski et al., 2016), a extensão *TS+PCA*, por meio de seus resultados obtidos, demonstra que o valor de k *shapelet* a ser selecionada é crucial para a predição, e que a quantidade seja um número médio, pois, não é suficiente apenas a redução do tamanho da série bem como selecionar as k *shapelet*, é fundamental agregar a elas valores característicos das séries que estejam ligadas diretamente a classe.

Em termo de acurácia, o algoritmo *Multilayer Perceptron* ocupa a primeira posição nesta extensão, com a média de 91,17%, na última posição está o algoritmo *Random Forest*

com a média de 80,30%. Conforme dito anteriormente é um ótimo resultado obtido em termo de desempenho dos modelos criados, pois a extensão é favorável mediante sua usabilidade no pré-processamento dos dados até a construção de classificadores.

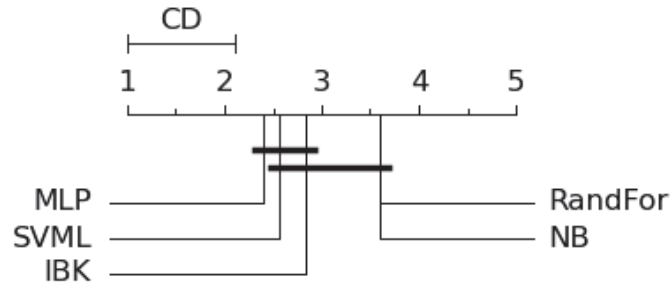
Tabela 5.3: Resultados dos experimentos realizados para a avaliação da extensão *TS+PCA*. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e o *rank* é a posição relativa do algoritmo em relação aos demais conjuntos de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	IBK	NB	SVMl	MLP	Rand For
1	98,47 (1)	96,68 (2)	55,50 (5)	70,08 (4)	96,42 (3)
2	73,33 (4,5)	73,33 (4,5)	80,00 (3)	86,67 (1,5)	86,67 (1,5)
3	98,22 (3)	99,33 (1)	98,89 (2)	98,11 (4)	95,78 (5)
4	97,40 (3)	78,70 (5)	91,67 (4)	99,53 (1)	98,54 (2)
5	100,00 (2)	96,43 (4,5)	100,00 (2)	96,43 (4,5)	100,00 (2)
6	87,26 (4)	83,99 (5)	97,71 (1)	90,52 (3)	90,85 (2)
7	88,00 (1)	79,00 (5)	82,00 (4)	87,00 (2)	86,00 (3)
8	99,65 (2)	99,42 (3)	99,88 (1)	99,19 (4)	97,56 (5)
9	77,28 (2)	73,67 (4,5)	77,52 (1)	74,26 (3)	73,67 (4,5)
10	95,46 (3)	77,27 (5)	98,86 (1,5)	98,86 (1,5)	85,23 (4)
11	87,12 (2)	82,73 (4)	79,56 (5)	88,54 (1)	86,98 (3)
12	98,29 (1,5)	97,14 (3)	98,29 (1,5)	96,57 (4,5)	96,57 (4,5)
13	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	96,67 (5)
14	95,24 (5)	95,72 (1,5)	95,72 (1,5)	95,63 (3,5)	95,63 (3,5)
15	59,02 (3,5)	59,02 (3,5)	60,66 (2)	62,30 (1)	57,38 (5)
16	79,45 (3)	76,71 (4)	82,19 (1,5)	82,19 (1,5)	72,60 (5)
17	81,71 (3)	50,79 (5)	74,87 (4)	91,58 (1)	86,84 (2)
18	92,09 (3)	93,69 (1)	93,05 (2)	90,90 (4)	84,90 (5)
19	93,33 (5)	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)
20	80,17 (3)	76,45 (5)	85,95 (1)	85,12 (2)	78,93 (4)
21	86,52 (3)	78,87 (5)	94,34 (2)	96,17 (1)	84,36 (4)
22	88,48 (2)	79,84 (5)	81,11 (4)	90,56 (1)	84,96 (3)
23	99,33 (2)	87,33 (5)	98,33 (3)	100,00 (1)	98,00 (4)
24	98,00 (4,5)	100,00 (1)	98,00 (4,5)	99,00 (2,5)	99,00 (2,5)
25	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	100,00 (2,5)	98,86 (5)
Média Acc	90,15	85,44	88,96	91,17	89,30
Rank Médio	2,84	3,60	2,56	2,40	3,60

Ainda nesta avaliação foi realizada uma análise estatística por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 5$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 5.3, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 3.58$ e $p\text{-valor} = 0.009137$. Desse modo, o resultado estatístico aponta diferença significativa existente entre os modelos. Para verificar quais são as diferenças significativas nos algoritmos nessa extensão, o pós-teste de *Naminye* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 1.11$.

A figura 5.5 representa o diagrama de diferença crítica existente entre os algoritmos nessa abordagem.

Figura 5.5: Diagrama de diferença crítica para o pós-teste de *Naminye* na extensão *TS+PCA*.



5.5 Considerações Finais

Neste capítulo foi apresentado uma descrição dos conjuntos de dados e o método *holdout* usado para a indução dos dados nos modelos de predição, também foi enfatizado a configuração dos experimentos realizados neste trabalho. Os resultados obtidos nesta seção evidenciam a capacidade das extensões da transformada *shapelet* em solucionar problemas de classificação de séries temporais. Mediante a análise estatística utilizada foi constatada diferenças críticas existentes entre os modelos preditivos nas extensões *TS+CS* e *TS+PCA* enquanto que na extensão *TS+CFS*, não houve diferenças críticas entre os algoritmos de classificação.

Portanto, no próximo capítulo será apresentada uma análise de comportamento individual de cada algoritmo de predição para cada extensão da transformada *shapelet* utilizada neste estudo.

Capítulo 6

Avaliação dos Algoritmos Preditivos

6.1 Considerações Iniciais

Neste capítulo é apresentada uma comparação de desempenho de cada algoritmo mediante a variação das extensões da transformada construídas. Consequentemente, a análise é válida para avaliação de cada conjunto de dados em relação ao algoritmo na extensão da *TS*.

Com base nos resultados de acurácias demonstrados anteriormente, foi possível construir uma análise exclusiva de avaliação de cada algoritmo por meio dos resultados obtidos, com o propósito de demonstrar qual das extensões da *TS* é mais adequada na utilização dos algoritmos preditivos. Portanto, são apresentadas as tabelas 6.1, 6.2, 6.3, 6.4 e 6.5, cujo os valores destacados em negrito representam os melhores resultados obtidos pelos algoritmos mediante as extensões em relação a cada conjuntos de dados. A partir dos resultados das tabelas acima referenciadas, foram construídas as figuras 6.1, 6.4, 6.6, 6.7 e 6.9, que ilustram as variações de comportamentos em termo de acurácia obtida em cada conjunto de dados pelos algoritmo através das extensões da transformada *shapelet* sobre os 25 conjuntos de dados exibidos no eixo do x em termo de acurácia representadas no eixo do y.

6.2 Avaliação do Algoritmo *IBK*

Os algoritmos baseados em distâncias, como exemplo as *ED* e as *DTW*, são algoritmos considerados estado-da-arte para classificação de séries temporais (Bagnall, 2012) e (Zalewski, 2015). Podem ser notados os resultados de acurácias apresentados pelo algoritmo *IBK* numa configuração padrão do classificador, usando apenas 1 vizinho mais próximo constata-se que seus resultados são difíceis de serem superados por outros modelos com princípios de inferências diferentes. Nesta análise, apresentamos a variação do algoritmo como consta na tabela 6.1 por intermédio das três extensões criadas (*TS+CFS*, *TS+CS* e *TS+PCA*).

A variação do algoritmo *IBK* (*TS+PCA*) registra desempenhos promissores em relação as duas outras abordagens (*IBK* (*TS+CFS*) e *IBK* (*TS+CS*)). Com acurácia média de 90,15% em *n* conjuntos de dados, o algoritmo aparece na primeira posição em vários conjuntos de dados, de acordo com o *rank médio* de 1,46 demonstrado na tabela 6.1. Portanto, destacamos dois fatores que influenciaram diretamente nesses resultados:

1. A seleção das melhores *shapelets* pela extensão *TS+PCA* é uma das mais eficientes abordagens utilizadas, pois, o modelo aufere de maneira certa a agregação de informações importantes e reduz o tamanho da série mostrando somente as *shapelets* discriminatórias.

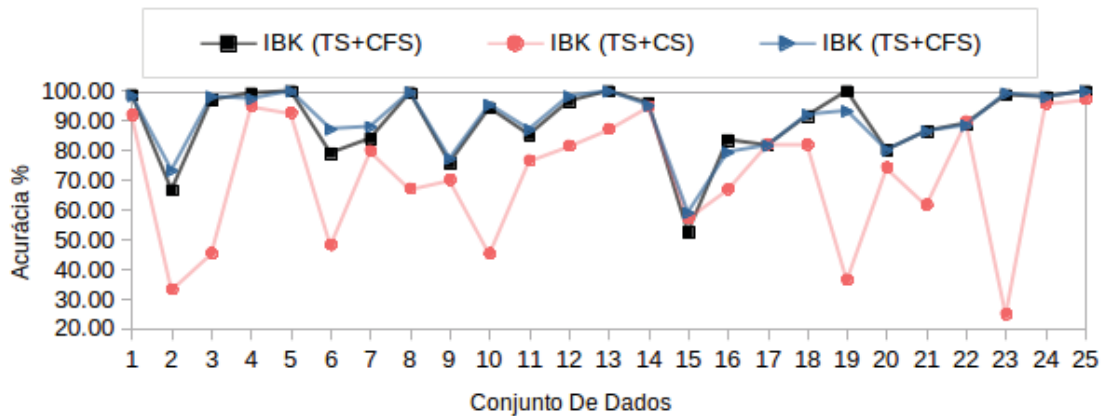
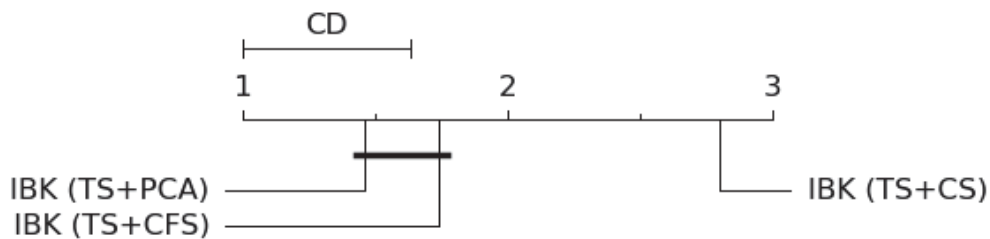
2. O fator de inferência pelo algoritmo de classificação é uma continuação do modelo de seleção de atributos pois as *shapelets* encontram-se ordenadas. O fator de similaridade foi intensificado na fase de pré-processamento, tornando o processo menos complexo para o algoritmo de indução.

Tabela 6.1: Resultados da variação do algoritmo (IBK) mediante as extensões TS+CFS, TS+CS e TS+PCA.

#	IBK (TS+CFS)	IBK (TS+CS)	IBK (TS+PCA)
1	98,72 (1)	92,07 (3)	98,47 (2)
2	66,67 (2)	33,33 (3)	73,33 (1)
3	97,00 (2)	45,56 (3)	98,22 (1)
4	99,25 (1)	95,16 (3)	97,40 (2)
5	100,00 (1,5)	92,86 (3)	100,00 (1,5)
6	79,09 (2)	48,37 (3)	87,26 (1)
7	84,00 (2)	80,00 (3)	88,00 (1)
8	99,42 (2)	67,25 (3)	99,65 (1)
9	75,80 (2)	70,24 (3)	77,28 (1)
10	94,32 (2)	45,46 (3)	95,46 (1)
11	85,27 (2)	76,78 (3)	87,12 (1)
12	96,57 (2)	81,71 (3)	98,29 (1)
13	100,00 (1,5)	87,33 (3)	100,00 (1,5)
14	96,02 (1)	94,95 (3)	95,24 (2)
15	52,46 (3)	57,38 (2)	59,02 (1)
16	83,56 (1)	67,12 (3)	79,45 (2)
17	81,71 (2,5)	82,24 (1)	81,71 (2,5)
18	91,69 (2)	82,11 (3)	92,09 (1)
19	100,00 (1)	36,67 (3)	93,33 (2)
20	80,17 (1,5)	74,38 (3)	80,17 (1,5)
21	86,52 (1,5)	61,90 (3)	86,52 (1,5)
22	89,12 (2)	89,92 (1)	88,48 (3)
23	98,67 (2)	25,03 (3)	99,33 (1)
24	98,00 (1,5)	96,00 (3)	98,00 (1,5)
25	100,00 (1,5)	97,45 (3)	100,00 (1,5)
Acc. Media	89,36	71,25	90,15
Rank Medio	1,74	2,80	1,46

Através da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 6.1, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 23.96$ e $p\text{-valor} = 0001$. Para verificar quais são as diferenças significativas no algoritmo mediante as extensões da transformada *shapelet*, o pós-teste de *Nemenyi* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0.66$.

Portanto, de acordo com a análise realizada, chegamos a conclusão de que a variação *IBK (TS+PCA)* é a mais adequada, e consequentemente, é a melhor nesta abordagem.

Figura 6.1: Variação do algoritmo *IBK* nas *ETS*.Figura 6.2: Diagrama de diferença crítica do algoritmo *IBK* nas *ETS*.

6.3 Avaliação do Algoritmo *Naive Bayes* (*NB*)

A avaliação do algoritmo *NB* também apresenta resultados de desempenhos favoráveis no âmbito do uso da extensão *TS+*. Qualquer análise mediante a predição dos resultados obtidos tem como fator primordial o método de seleção das *shapelets* que influenciam diretamente nos resultados. Contudo, para este modelo preditivo, a classificação ocorre combinando a probabilidade de várias hipóteses a serem mensuradas. Uma vez que as *shapelets* encontram-se ordenadas por meio de alguma medida de importância entre elas, o método probabilístico beneficia-se desta vantagem e reduz seu custo computacional, ou seja, não explora probabilidades que não vão de acordo com a representação dos dados induzidos, diminuindo o tempo de predição. Neste cenário, as variações demonstram certa consistência mediante este algoritmo. Quanto aos valores com 87,74% de acurácia média e um *rank* médio de 1.64, a variação *NB* (*TS+CFS*) encontra-se em primeiro lugar se comparada com as variações *NB* (*TS+PCA*) e *NB* (*TS+CS*).

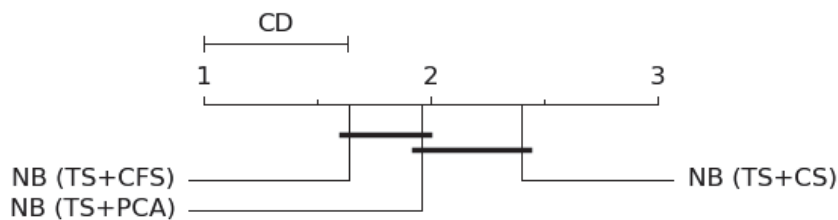
O algoritmo explora o máximo possível todas as informações disponíveis de subsequências construídas pelos modelos de seleção das *shapelets* na fase de pré-processamento dos dados. Ao notarmos os resultados descrito na tabela 6.2, cumpre-se um dos nossos objetivos que é o de estendermos a transformada *shapelet* e representar os dados em outro domínio, podendo com isso, haver maiores ganhos de informações, bem como melhorar o desempenho do algoritmo de classificação sem a necessidade de ajustes de parâmetros.

Através da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 6.2, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 4.0$ e $p\text{-valor} = 0.024096$. Para verificar quais são as diferenças significativas no algoritmo *Naive Bayes* (*NB*) mediante as extensões da transformada *shapelet*, o pós-teste de *Nemenyi* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0.66$.

Tabela 6.2: Resultados da variação do algoritmo *Naive Bayes* (NB) mediante as extensões *TS+CFS*, *TS+CS* e *TS+PCA*.

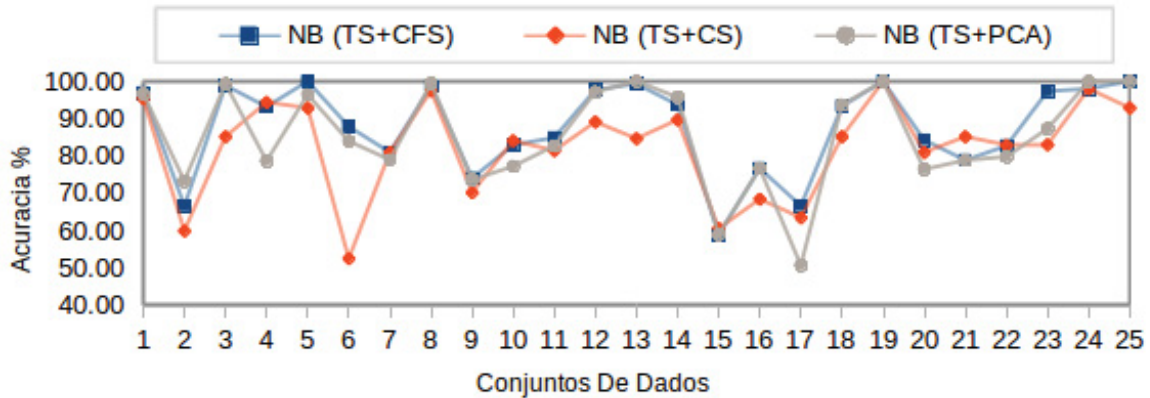
#	NB (TS+CFS)	NB (TS+CS)	NB (TS+PCA)
1	96,93 (1)	95,40 (3)	96,68 (2)
2	66,67 (2)	60,00 (3)	73,33 (1)
3	98,89 (2)	85,22 (3)	99,33 (1)
4	93,18 (2)	94,38 (1)	78,70 (3)
5	100,00 (1)	92,86 (3)	96,43 (2)
6	87,91 (1)	52,61 (3)	83,99 (2)
7	81,00 (1,5)	81,00 (1,5)	79,00 (3)
8	98,72 (2)	97,45 (3)	99,42 (1)
9	74,08 (1)	70,24 (3)	73,67 (2)
10	82,96 (2)	84,09 (1)	77,27 (3)
11	84,93 (1)	81,46 (3)	82,73 (2)
12	97,71 (1)	89,14 (3)	97,14 (2)
13	99,33 (2)	84,67 (3)	100,00 (1)
14	93,68 (2)	89,70 (3)	95,72 (1)
15	59,02 (2,5)	60,66 (1)	59,02 (2,5)
16	76,71 (1,5)	68,49 (3)	76,71 (1,5)
17	66,71 (1)	63,55 (2)	50,79 (3)
18	93,29 (2)	85,22 (3)	93,69 (1)
19	100,00 (2)	100,00 (2)	100,00 (2)
20	84,30 (1)	80,99 (2)	76,45 (3)
21	78,87 (2,5)	85,19 (1)	78,87 (2,5)
22	82,72 (2)	82,88 (1)	79,84 (3)
23	97,33 (1)	83,04 (3)	87,33 (2)
24	98,00 (2,5)	98,00 (2,5)	100,00 (1)
25	100,00 (1,5)	92,89 (3)	100,00 (1,5)
<i>Acc. Media</i>	87,72	82,37	85,44
<i>Rank Medio</i>	1,64	2,4	1,96

Figura 6.3: Diferença crítica da variação do algoritmo *NB* mediante as extensões da *TS*.



A figura 6.3 apresenta a diferença estatística existente entre as variações. A formação de dois *cliques* que agregam as mesmas variações traduzem a inexistência de diferenças críticas entre elas, fazendo com que a variação *NB (TS+CFS)* seja de fato a melhor abordagem a ser empregada nessa avaliação através da análise do algoritmo (*NB*).

Figura 6.4: Variação do algoritmo *Naive Bayes* (NB) nas extensões da TS.



6.4 Avaliação do Algoritmo SVM

As *SVML* (*máquinas de vetores de suportes lineares*), buscam por um hiperplano ótimo, com o objetivo de maximizar as distâncias entre os exemplos semelhantes a uma classe e minimizar a distância dos exemplos da mesma classe (Cortes C, 1995). Nós usamos o *SVML* pois já existem resultados da transformada *shapelet* extraídos em Hills et al. (2014), que podem ser comparados com as extensões que construídas. Conforme mencionado anteriormente, o modelo de seleção de atributos já realizou a separação das melhores *shapelets* e efetuou uma ordenação, isso facilita o processo do modelo de predição das *SVML* a ser menos trivial. Conforme a tabela 6.3, os resultados desta variação são apresentados em termo de acurácia. Em cada conjunto de dados pode ser notado o quanto o modelo é ou não funcional para ser utilizado nas extensões da TS. Quanto ao comportamento deste algoritmo nas variações das extensões com percentual de 88,96% e um *rank médio* de 7,5, a extensão *SVML (TS+PCA)* torna-se a mais adequada para ser utilizada em SVM.

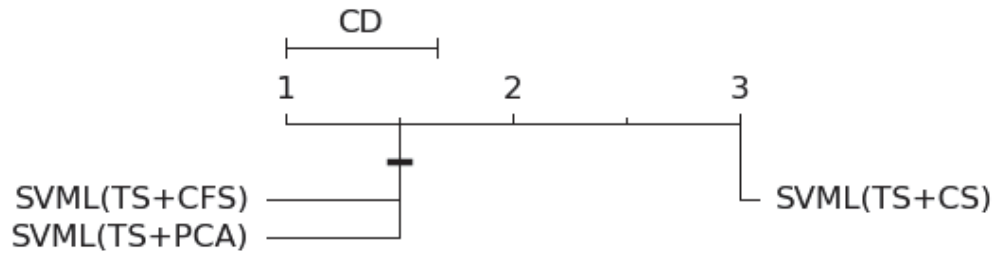
Continuando com a análise da variação desse algoritmo nas extensões da TS, por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank médio* descritos na tabela 6.4, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 72.0$ e $p\text{-valor} = 00001$. Para verificar quais são as diferenças significativas no algoritmo *SVML*, mediante as extensões da transformada *shapelet*, o pós-teste de *Nemenyi* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0.66$.

A figura 6.5 destaca fortemente tal diferença, uma vez que a variação *SVML (TS+CS)* não faz parte do *clique* *SVML (TS+PCA)* e *SVML (TS+CSF)*. De acordo com a análise inicial, estas variações acabam demonstrando equivalências de valores entre si, ou seja, qualquer uma das duas variações (*SVML (TS+PCA)* e *SVML (TS+CFS)*) nesta abordagem pode prover melhores resultados em termo de acurácias como mostra a figura 6.6.

6.5 Avaliação do Algoritmo MLP

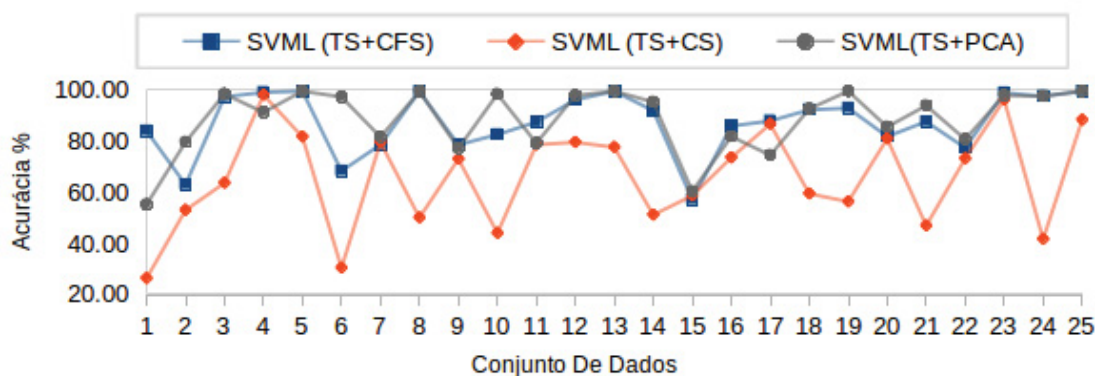
O algoritmo *Multilayer Perceptron* (MLP) em sua configuração padrão alcança o nível da consistência das extensões da transformada *shapelet* em termo de acurácia. É um classificador

Figura 6.5: Diagrama de diferença crítica do algoritmo SVML nas extensões da TS.

Tabela 6.3: Resultados da variação do algoritmo Máquinas de Vetores de Suportes Lineares (SVML) mediante as extensões *TS+CFS*, *TS+CS* e *TS+PCA*.

#	SVML (TS+CFS)	SVML (TS+CS)	SVML(TS+PCA)
1	84,40 (1)	26,59 (3)	55,50 (2)
2	63,33 (2)	53,33 (3)	80,00 (1)
3	97,78 (2)	64,00 (3)	98,89 (1)
4	99,51 (1)	98,54 (2)	91,67 (3)
5	100,00 (1,5)	82,14 (3)	100,00 (1,5)
6	68,63 (2)	30,72 (3)	97,71 (1)
7	79,00 (3)	80,00 (2)	82,00 (1)
8	99,88 (1,5)	50,41 (3)	99,88 (1,5)
9	78,94 (1)	73,43 (3)	77,52 (2)
10	82,96 (2)	44,32 (3)	98,86 (1)
11	87,85 (1)	79,02 (3)	79,56 (2)
12	96,57 (2)	80,00 (3)	98,29 (1)
13	100,00 (1,5)	78,00 (3)	100,00 (1,5)
14	92,32 (2)	51,51 (3)	95,72 (1)
15	57,38 (3)	59,02 (2)	60,66 (1)
16	86,30 (1)	73,97 (3)	82,19 (2)
17	88,29 (1)	87,11 (2)	74,87 (3)
18	92,65 (2)	59,82 (3)	93,05 (1)
19	93,33 (2)	56,67 (3)	100,00 (1)
20	82,23 (2)	81,41 (3)	85,95 (1)
21	88,02 (2)	47,26 (3)	94,34 (1)
22	77,76 (2)	73,60 (3)	81,11 (1)
23	99,33 (1)	96,67 (3)	98,33 (2)
24	98,00 (1,5)	42,00 (3)	98,00 (1,5)
25	100,00 (1,5)	88,76 (3)	100,00 (1,5)
Acc. Media	87,78	66,33	88,96
Rank Medio	1,5	3	1,5

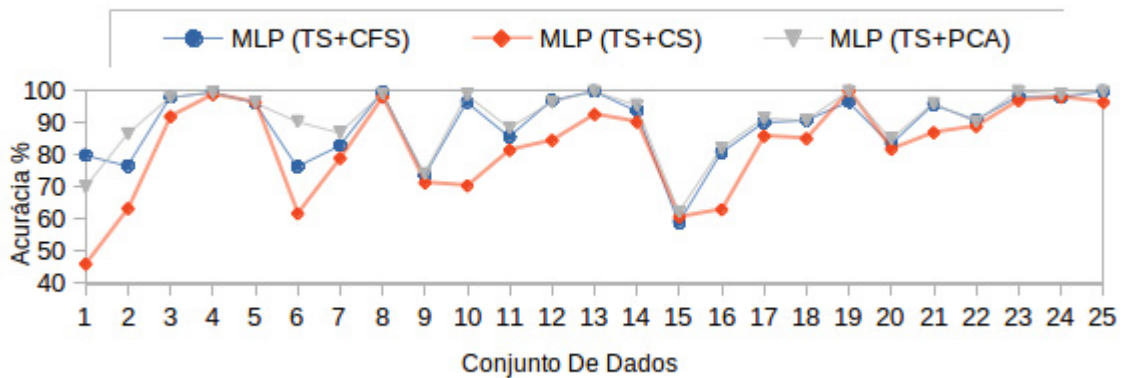
com um modelo matemático de inferência caixa preta, pois o modo de predição dos dados é menos inteligível em comparação com os algoritmos baseados em árvores de decisão e as *SVM*. No treinamento da rede, usualmente tem sido recomendado o algoritmo *Backpropagation*, e nesse estudo não foi diferente, porém, evitamos apenas os reajustes de parâmetros na rede. Com resultados médios de acurácia de 91,17% e um rank médio de 1,5 a variação *MLP (TS+PCA)* acaba sendo a abordagem mais adequada a ser utilizada neste algoritmo.

Figura 6.6: Variação do algoritmo *SVML* nas extensões da *TS*.Tabela 6.4: Resultados da variação do algoritmo *Multilayer Perceptron* (MLP) mediante as extensões TS+CFS, TS+CS e TS+PCA.

#	<i>MLP</i> (TS+CFS)	<i>MLP</i> (TS+CS)	<i>MLP</i> (TS+PCA)
1	80,05 (1)	46,04 (3)	70,08 (2)
2	76,67 (2)	63,33 (3)	86,67 (1)
3	98,00 (2)	92,00 (3)	98,11 (1)
4	99,69 (1)	98,85 (3)	99,53 (2)
5	96,43 (2)	96,43 (2)	96,43 (2)
6	76,47 (2)	61,77 (3)	90,52 (1)
7	83,00 (2)	79,00 (3)	87,00 (1)
8	99,65 (1)	98,03 (3)	99,19 (2)
9	73,61 (2)	71,36 (3)	74,26 (1)
10	96,59 (2)	70,46 (3)	98,86 (1)
11	85,85 (2)	81,56 (3)	88,54 (1)
12	97,14 (1)	84,57 (3)	96,57 (2)
13	100,00 (1,5)	92,67 (3)	100,00 (1,5)
14	93,88 (2)	90,28 (3)	95,63 (1)
15	59,02 (3)	60,66 (2)	62,30 (1)
16	80,82 (2)	63,01 (3)	82,19 (1)
17	90,26 (2)	85,92 (3)	91,58 (1)
18	90,9 (1,5)	85,14 (3)	90,9 (1,5)
19	96,67 (3)	100,00 (1,5)	100,00 (1,5)
20	83,47 (2)	81,82 (3)	85,12 (1)
21	95,84 (2)	87,02 (3)	96,17 (1)
22	91,04 (1)	88,96 (3)	90,56 (2)
23	98,33 (2)	97,00 (3)	100,00 (1)
24	98,00 (2,5)	98,00 (2,5)	99,00 (1)
25	100,00 (1,5)	96,49 (3)	100,00 (1,5)
<i>Acc. Media</i>	89,66	82,81	91,17
<i>Rank Medio</i>	1,5	3,0	1,5

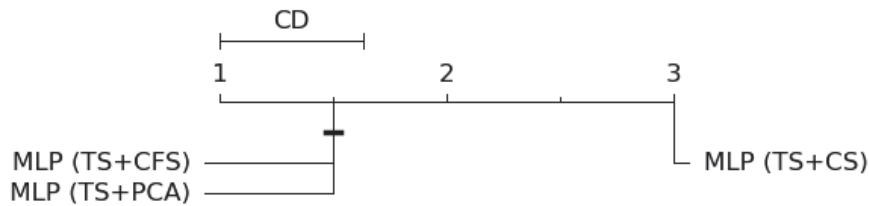
Ainda nesta análise, por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 6.4, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 72.0$ e $p\text{-valor} = 0.00001$. Para verificar quais são as diferenças significativas no algoritmo *MLP*, através das extensões da transformada *shapelet*, o pós-teste de *Nemenyi* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0.66$.

Figura 6.7: Variação do algoritmo *Multilayer Perceptron* (*MLP*) nas extensões da TS



A figura 6.8 ilustra de maneira precisa, onde a variação *MLP (TS+CS)* não faz parte do clique *MLP (TS+CPA)*, indicando uma grande diferença existente entre as variações. As variações *MLP (TS+CFS)* e *MLP (TS+CPA)* apresentam equivalência de eficiência em seus resultados, excluindo a existência de diferenças críticas entre elas, assim, qualquer uma das abordagens sustenta os conceitos da extensão da transformada *shapelet*.

Figura 6.8: Diferença crítica da variação (*MLP (TS+CFS)*, *MLP (TS+CS)* e *MLP (TS+PCA)*).



6.6 Avaliação do Algoritmo *Random Forest*

O algoritmo *Random Forest* é considerado um clássico no que tange aos modelos baseados em árvore de decisão, o algoritmo constrói várias árvores em sua fase de treinamento e gera as classes para a predição (Ho, 1995), é um algoritmo que na sua essência contém auto poder de generalização dos dados. Em 25 conjuntos de dados experimentados, as variações do algoritmo apresentaram resultados de acurácias favoráveis conforme descreve a tabela 6.5. Nota-se que a extensão *RanFor (TS+CS)* que sempre apareceu nos últimos lugares mediante as

variações da *TS+*, nesse algoritmo, o modelo aparece com seus resultados mais competitivos chegando a ocupar os primeiros lugares em alguns conjuntos de dados, o que intensifica a consistência da transformada *shapelet*. Apesar desta competitividade, destaca-se o desempenho da variação *RandFor (TS+CFS)* com valores de acurácia média de 89,39% e um rank médio de 1,34.

Tabela 6.5: Resultados da variação do algoritmo Random Forest (*RandFor*) mediante as extensões *TS+CFS*, *TS+CS* e *TS+PCA*.

#	<i>RandFor (TS+CFS)</i>	<i>RandFor (TS+CS)</i>	<i>RandFor (TS+PCA)</i>
1	98,47 (1)	97,44 (2)	96,42 (3)
2	80,00 (2)	63,33 (3)	86,67 (1)
3	94,56 (2)	93,00 (3)	95,78 (1)
4	99,82 (1)	99,01 (2)	98,54 (3)
5	100,00 (1,5)	96,43 (3)	100,00 (1,5)
6	75,49 (2)	61,11 (3)	90,85 (1)
7	82,00 (2,5)	82,00 (2,5)	86,00 (1)
8	97,45 (3)	98,49 (1)	97,56 (2)
9	76,69 (1)	72,96 (3)	73,67 (2)
10	90,91 (1)	80,68 (3)	85,23 (2)
11	85,85 (2)	82,83 (3)	86,98 (1)
12	97,71 (1)	90,29 (3)	96,57 (2)
13	98,00 (1)	92,67 (3)	96,67 (2)
14	89,99 (2)	89,12 (3)	95,63 (1)
15	63,93 (1)	54,10 (3)	57,38 (2)
16	75,34 (1)	71,23 (3)	72,60 (2)
17	89,61 (1)	88,68 (2)	86,84 (3)
18	86,34 (1)	85,14 (2)	84,90 (3)
19	100,00 (2)	100,00 (2)	100,00 (2)
20	83,47 (1)	76,03 (3)	78,93 (2)
21	88,69 (1)	86,02 (2)	84,36 (3)
22	86,40 (2)	87,36 (1)	84,96 (3)
23	97,00 (2)	95,67 (3)	98,00 (1)
24	98,00 (2,5)	98,00 (2,5)	99,00 (1)
25	99,03 (1)	96,75 (3)	98,86 (2)
<i>Acc. Media</i>	89,39	85,53	89,30
<i>Rank Medio</i>	1,54	2,56	1,9

Na análise estatística realizada por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 6.5, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 8,78$ e $p\text{-valor} = 0,00056$. Para verificar quais são as diferenças significativas no algoritmo *Random Forest*, de acordo com as extensões da transformada *shapelet*, o pós-teste de *Naminye* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 0,66$.

A figura 6.10 demonstra a diferença crítica existente entre os modelos de classificação mediante o diagrama CD. Portanto, nesta abordagem, o algoritmo ou avariação *RandFor (TS+CFS)* é a melhor opção para representar esta extensão.

Figura 6.9: Variação do algoritmo *Random Forest* nas extensões da TS.

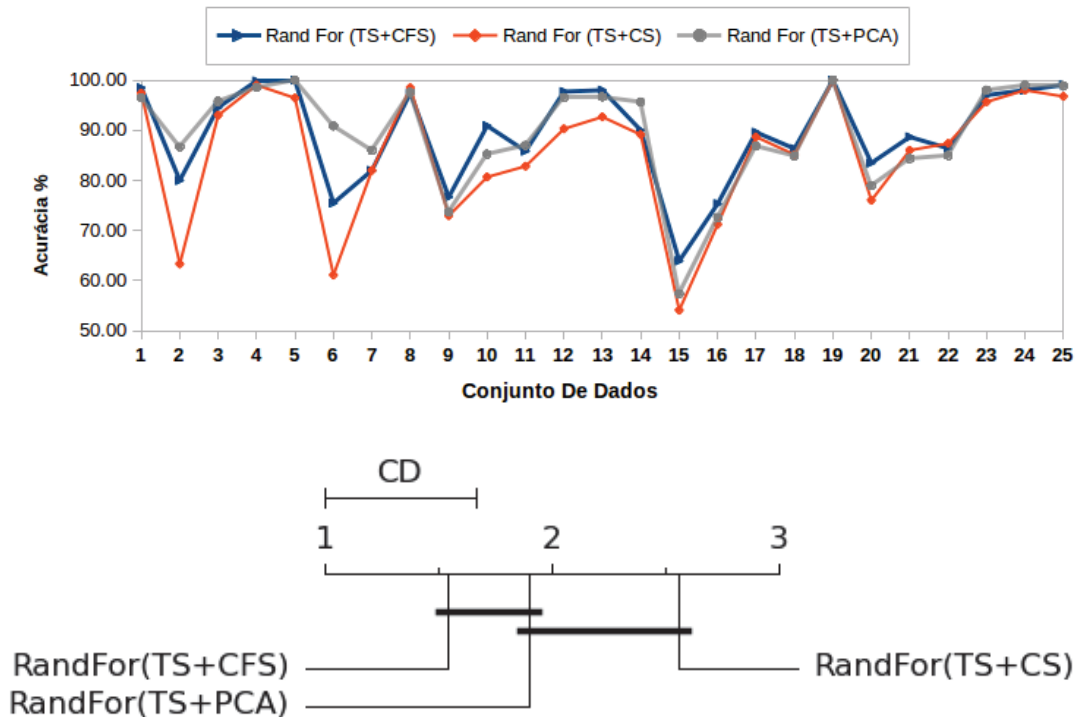


Figura 6.10: Diagrama de diferença crítica para o pós-teste de *Nemenyi* no algoritmo *Random Forest* (*RandFor*).

6.7 Avaliação do Algoritmo J48

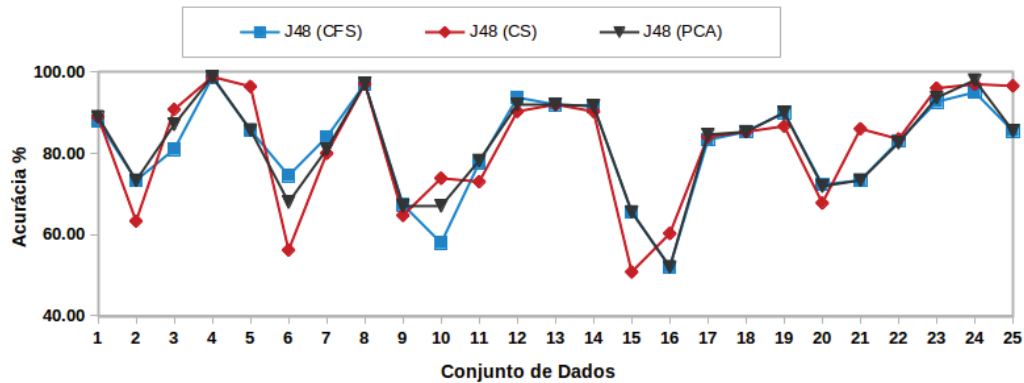
Inicialmente na proposta original da transformada *shapelet* (Lines et al., 2012) e (Zalewski et al., 2016), foi utilizado o algoritmo *J48* para a criação de árvore de decisão com o propósito de predição das séries temporais. Este algoritmo faz parte da família dos modelos preditivos baseados em procura, sendo um algoritmo robusto em termos de processamentos e inferências de dados (Faceli et al., 2011). Conforme os experimentos realizados, a tabela 6.6 apresenta as variações do algoritmo *J48* nas extensões *TS+CFS*, *TS+CS* e *TS+PCA*, e com base nestes resultados nota-se a superioridade da variação do algoritmo na extensão (*J48(TS+PCA)*) com média geral de 82,03% de acurácia em n conjuntos de dados. A figura 6.11 ilustra a variação do algoritmo *J48* em relação aos 25 conjunto de dados.

Por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 3$ e $N = 25$ sobre os valores de *rank* médio descritos na tabela 6.6, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 0,36$ e $p\text{-valor} = 0,699$, mas os valores do $p\text{-valor}$ indicam o contrário. Nesse caso não foi possível rejeitar a hipótese nula, portanto não há diferença estatística significativa nas variações do algoritmo *J48* nas extensões da transformada *shapelet*.

6.8 Comparação dos Melhores Algoritmos

Esta avaliação enfatiza a comparação dos melhores algoritmos de classificação adotados mediante as extensões da transformada *shapelet* em termos de desempenhos. Conforme comentado anteriormente, no desempenho das extensão da *TS* foi escolhido um algoritmo

Figura 6.11: Variação do algoritmo J48



como sendo a melhor opção para utilização daquela extensão. Portanto, extraímos os melhores algoritmos com o propósito de comparar e verificar a existência de diferenças estatísticas entre eles.

A tabela 6.7 descreve as melhores variações dos algoritmos nas extensões da TS mediante os n conjuntos de dados. As últimas duas linhas representam para cada algoritmo os valores de acurácia média e *rank médio* respectivamente. As células destacadas representam os melhores valores de acurácia para um determinado conjunto de dados considerando todos os algoritmos avaliados.

Na discussão dos resultados dos melhores algoritmos, em cada extensão foi realizada uma análise estatística por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 5$ e $N = 25$ sobre os valores de *rank médio* descritos na tabela 6.6, a hipótese nula de que todos os modelos são equivalentes, em termos de acurácia foi rejeitada com valores de $F_F = 1,18$ e $p\text{-valor} = 0,32$, mas o valor de $p\text{-valor}$ indica o contrário. Nesse caso não foi possível rejeitar a hipótese nula, portanto, não há diferença estatística significativa nas variações dos melhores algoritmos nas extensões da transformada *shapelet*.

6.9 Considerações Finais

Neste capítulo foi demonstrada uma análise individual dos modelos de classificação utilizados nas extensões da transformada *shapelet*. Por meio desta análise os resultados apontavam para a eficiência das extensões da transformada *shapelet* em termo de acurácia, como se esperava. Assim, mediante as análises estatísticas, os resultados apontaram para equivalências destes algoritmos. Neste contexto, para cada extensão da transformada, nossas pesquisas recomendam a utilização destes modelos nestas mesmas configurações. No próximo capítulo será realizada uma comparação da transformada *shapelet* proposta por Hills et al. (2014) através da sua mais recente técnica de busca das melhores k *shapelets* em relação aos métodos característicos das extensões da transformada *shapelet* discutidos neste estudo.

Tabela 6.6: Resultados do algoritmo J48 baseados nas extensões da TS. Os valores estão representados em *acurácia* e um *rank* atribuído a cada variação em relação as demais variação do algoritmo respectivamente. Os valores destacados em negrito são os melhores valores de acurácia em cada conjunto de dados em relação a variação do algoritmo.

#	<i>J48 (CFS)</i>	<i>J48 (CS)</i>	<i>J48 (PCA)</i>
1	87,98 (3)	89,00 (1,5)	89,00 (1,5)
2	73,33 (1,5)	63,33 (3)	73,33 (1,5)
3	81,00 (3)	90,78 (1)	87,11 (2)
4	98,67 (3)	98,80 (2)	98,88 (1)
5	85,71 (2,5)	96,43 (1)	85,71 (2,5)
6	74,51 (1)	56,21 (3)	67,97 (2)
7	84,00 (1)	80,00 (3)	81,00 (2)
8	97,10 (2)	97,10 (2)	97,10 (2)
9	67,34 (1)	64,73 (3)	67,04 (2)
10	57,96 (3)	73,86 (1)	67,05 (2)
11	77,66 (2)	73,02 (3)	78,24 (1)
12	93,71 (1)	90,29 (3)	92,00 (2)
13	92,00 (2)	92,00 (2)	92,00 (2)
14	91,64 (1,5)	90,28 (3)	91,64 (1,5)
15	65,57 (1,5)	50,82 (3)	65,57 (1,5)
16	52,06 (2,5)	60,27 (1)	52,06 (2,5)
17	83,29 (3)	84,08 (2)	84,61 (1)
18	85,30 (2)	85,30 (2)	85,30 (1,5)
19	90,00 (1,5)	86,67 (3)	90,00 (1,5)
20	72,31 (1)	67,77 (3)	71,90 (2)
21	73,38 (2,5)	86,02 (1)	73,38 (2,5)
22	83,04 (2)	83,52 (1)	82,56 (2)
23	92,67 (3)	6,00 (2)	93,67 (1)
24	95,00 (3)	97,00 (2)	98,00 (1)
25	85,51 (2,5)	96,58 (1)	85,51 (2)
<i>Acc. Média</i>	81,63	81,99	82,03
<i>Rank Médio</i>	2,08	2,06	1,86

Tabela 6.7: Uma abordagem comparativa de avaliação das melhores variações dos algoritmos nas extensões da transformada *shapelet* sobre os 25 conjuntos de dados. Os valores em negrito representam a melhor acurácia obtida na variação dos conjuntos de dados.

	IBK (TS+PCA)	NB (TS+CFS)	MLP (TS+CFS)	SVML(TS+PCA)	RandFo (TS+PCA)
1	98,47 (1)	96,93 (2)	80,05 (4)	55,5 (5)	96,42 (3)
2	73,33 (4)	66,67 (5)	76,67 (3)	80 (2)	86,67 (1)
3	98,22 (3)	98,89 (1,5)	98 (4)	98,89 (1,5)	95,78 (5)
4	97,4 (3)	93,18 (4)	99,69 (1)	91,67 (5)	98,54 (2)
5	100 (2,5)	100 (2,5)	96,43 (5)	100 (2,5)	100 (2,5)
6	87,26 (4)	87,91 (3)	76,47 (5)	97,71 (1)	90,85 (2)
7	88,00 (1)	81,00 (5)	83,00 (3)	82,00 (4)	86,00 (2)
8	99,65 (2,5)	98,72 (4)	99,65 (2,5)	99,88 (1)	97,56 (5)
9	77,28 (2)	74,08 (3)	73,61 (5)	77,52 (1)	73,67 (4)
10	95,46 (3)	82,96 (5)	96,59 (2)	98,86 (1)	85,23 (4)
11	87,12 (1)	84,93 (4)	85,85 (3)	79,56 (5)	86,98 (2)
12	98,29 (1,5)	97,71 (3)	97,14 (4)	98,29 (1,5)	96,57 (5)
13	100 (2)	99,33 (4)	100 (2)	100 (2)	96,67 (5)
14	95,24 (3)	93,68 (5)	93,88 (4)	95,72 (1)	95,63 (2)
15	59,02 (3)	59,02 (3)	59,02 (3)	60,66 (1)	57,38 (5)
16	79,45 (3)	76,71 (4)	80,82 (2)	82,19 (1)	72,6 (5)
17	81,71 (3)	66,71 (5)	90,26 (1)	74,87 (4)	86,84 (2)
18	92,09 (3)	93,29 (1)	90,9 (4)	93,05 (2)	84,9 (5)
19	93,33 (5)	100 (2)	96,67 (4)	100 (2)	100 (2)
20	80,17 (4)	84,3 (2)	83,47 (3)	85,95 (1)	78,93 (5)
21	86,52 (3)	78,87 (5)	95,84 (1)	94,34 (2)	84,36 (4)
22	88,48 (2)	82,72 (4)	91,04 (1)	81,11 (5)	84,96 (3)
23	99,33 (1)	97,33 (5)	98,33 (2,5)	98,33 (2,5)	98 (4)
24	98 (3,5)	98 (3,5)	98 (3,5)	98 (3,5)	99 (1)
25	100 (2,5)	100 (2,5)	100 (2,5)	100 (2,5)	98,86 (5)
acc Media	90,15	87,71	89,65	88,96	89,29
rank medio	2,66	3,52	3	2,4	3,42

Capítulo 7

Comparação da *TS* Versus as *ETS*

7.1 Considerações Iniciais

Na transformada *shapelet* proposta por (Lines et al., 2012) e posteriormente melhorada nas pesquisas de (Hills et al., 2014), o processo de busca das melhores *shapelets* é intensificado pelas estimativas manuais dos parâmetros máximos e mínimos das subsequências possíveis da série para encontrar as melhores *shapelets*. Em Hills et al. (2014), os autores propõem a metodologia de divisão do tamanho da série temporal por dois ($m/2$), onde o valor resultante é atribuído a k que constitui as melhores *shapelets*. A metodologia também apresenta ótimos resultados em termo de acurácia, porém, o número elevado de *shapelets* utilizadas para indução influencia diretamente no custo computacional.

7.2 Avaliação de Resultados

As tabelas 7.1 e 7.2 descrevem os valores de resultados de uma maneira comparativa da transformada *shapelet TS* ($m/2$) em relação as extensões da transformada *shapelet* (*TS+CS*, *TS+CFS* e *TS+PCA*) em termos de desempenhos, nesta avaliação, consideramos apenas dezesseis (16) dos 25 conjuntos de dados testados em todas as abordagens mencionadas anteriormente. Nessa comparação, pretendemos mostrar que as extensões da transformada *shapelet* (*TS*) apesar de serem constituídas por um número reduzido de *shapelet*, apresentaram melhores resultados de desempenhos em comparação com a transformada *shapelet TS* original, de acordo com os resultados de Hills et al. (2014).

Em termo de acurácia a *TS* original apresenta desempenhos inferiores em relação as extensões *TS+CFS* e *TS+PCA*, exceto pela *TS+CS*, cujo os valores de desempenhos são inferiores em relação as demais extensões. Na mesma tabela é possível notar os *ranks* atribuídos nas abordagens, onde as primeiras posições são ocupadas pelas extensões *TS+CFS* e *TS+PCA* e algumas pela *TS+(m/2)*. Uma das análises que pode ser mensurada é que nos conjuntos de dados (*Coffe*, *FaceFour*, *GunPoint* e *Trace*), a transformada *shapelet TS m/2* proposta por (Hills et al., 2014), apresenta valores de acurácia de 99,9% em alguns conjuntos de dados, e as extensões propostas neste estudo também atingem os mesmos resultados de desempenhos. Isso demonstra o quanto as extensões são eficientes em selecionar as melhores k *shapelets* presentes nos conjuntos de dados e que podem ser utilizadas na indução. Os resultados cumprem os objetivos desta pesquisa pois equiparam a transformada *shapelet* e suas extensões em termos de desempenhos.

De acordo com os resultados apresentados nas tabelas 7.1 e 7.2 foi possível construir as figuras 7.1 e 7.2, essas definem uma comparação da variação dos algoritmos na transformada

shapelet especificamente no método proposto por Hills et al. (2014), em relação as extensões da transformada reduzida (*TS+*). A comparação é baseada no mesmo algoritmo com base nas duas abordagens. A figura 7.1 descreve a variação comparativa do algoritmo *SVML* por meio do método de seleção das melhores *shapelets*, enquanto que na figura 7.2 se destaca a comparação do algoritmo *Random Forest* nas duas abordagens, lembrando que os resultados da transformada *shapelet* (*TS m/2*) foram extraídos em Hills et al. (2014). Nesta avaliação conclui-se que as variações dos algoritmos aqui comparados nas extensões da *TS*, apresentam alternativas de maiores valores de acurácia em relação ao método da transformada *shapelet* (*TS m/2*).

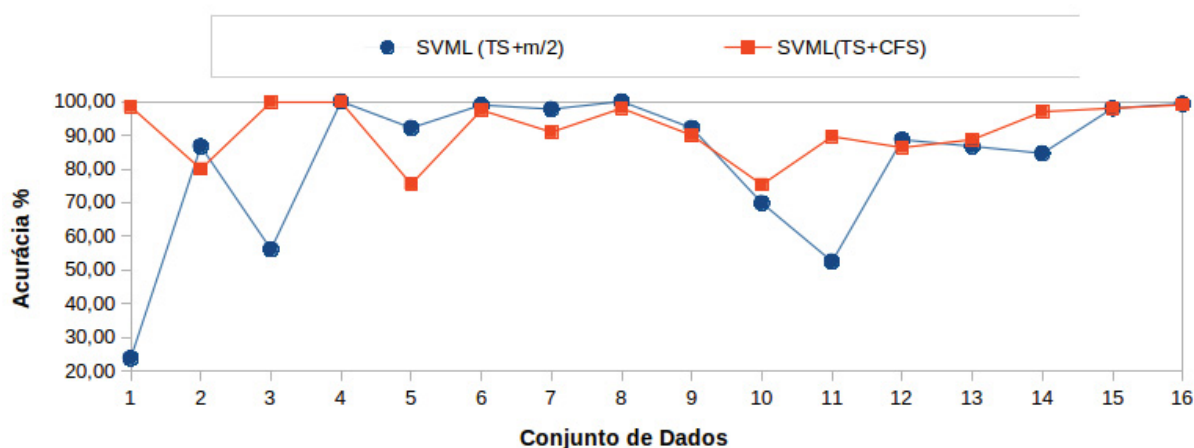


Figura 7.1: Comparação do algoritmo *SVML* na transformada *TS(m/2)* Versus o mesmo algoritmo na extensão da transformada *shapelet* *SVML (TS+PCA)*.

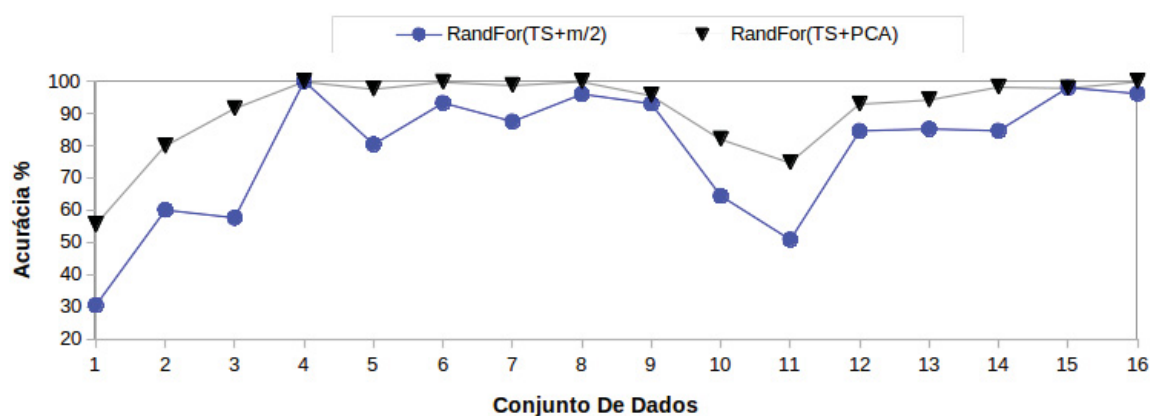


Figura 7.2: Comparação do algoritmo *Random Forest* na transformada *TS+(m/2)* Versus *Random Forest (TS+PCA)*.

Os valores de acurácia média nos algoritmos *SVML (TS +m/2)*, *SVML (TS+PCA)*, *RandoFor (TS m/2)* e *RandoFor (TS+PCA)*, para os 16 conjuntos foram de 82,95%, 92,01%, 78,87% e 91,26% respectivamente. Nesta avaliação o algoritmo *Random Forest* na extensão *TS+PCA* apresentou os melhores resultados de acurácia em 9 conjuntos de dados em relação aos demais algoritmos. Esta avaliação também pode ser constatada mediante a descrição da tabela 7.1 e 7.2.

Tabela 7.1: Resultados dos experimentos realizados para a avaliação 3, abordagem comparativa em relação a transformada *shapelet*. Os valores são apresentados no formato (*acc rank*), no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados. Os resultados da abordagem *TS+m/2* foram extraídos de Hills et al. (2014).

#	<i>TS+m/2</i>			<i>TS+CFS</i>		
	<i>B. Network</i>	<i>R. Forest</i>	<i>SVM (linear)</i>	<i>B. network</i>	<i>R. Forest</i>	<i>SVM (linear)</i>
1	25.00 (11)	30.43 (9)	23.79 (12)	96,16 (4)	84,40 (7)	98,47 (1)
2	90.00 (1)	60.00 (11)	86.67 (2)	83,33 (5)	63,33 (9,5)	80,00 (6,5)
4	57.08 (11)	57.58 (10)	56.15 (12)	96,12 (7)	99,51 (2)	99,82 (1)
5	96.43 (7)	100.00 (3,5)	100.00 (3,5)	96,43 (9,5)	100,00 (3,5)	100,00 (3,5)
6	90.2 (4)	80.39 (6)	92.16 (2)	66,67 (9)	68,63 (8)	75,49 (7)
8	99.54 (4)	93.26 (11)	98.95 (6)	99,07 (5)	99,88 (1,5)	97,45 (10)
10	100 (1)	87.5 (7)	97.73 (3)	93,18 (4)	82,96 (9)	90,91 (6)
13	99.33 (6)	96 (9)	100.00 (2,5)	99,33 (5)	100,00 (2,5)	98,00 (7)
14	92.42 (5)	93 (4)	92.13 (7)	89,70 (10)	92,32 (6)	89,99 (9)
16	65.75 (11)	64.38 (12)	69.86 (10)	79,45 (4)	86,30 (1)	75,34 (5)
17	28.16 (12)	50.79 (11)	52.5 (10)	84,47 (6)	88,29 (3)	89,61 (1)
18	89.06 (5)	84.58 (11)	88.66 (6)	93,61 (2)	92,65 (4)	86,34 (7)
21	89.68 (4)	85.19 (10)	86.69 (7)	94,51 (1,5)	88,02 (6)	88,69 (5)
23	92.26 (8)	84.62 (9,5)	84.62 (9,5)	95,33 (7)	99,33 (1)	97,00 (4)
24	100.00 (1,5)	98.00 (7,5)	98.00 (7,5)	98,00 (7,5)	98,00 (7,5)	98,00 (7,5)
25	98.77 (8)	96.14 (11)	99.3 (5)	100,00 (2,5)	100,00 (2,5)	99,03 (6)
Média	82,11	78,87	82,95	91,59	90,23	91,51
Rank Médio	6,22	8,91	6,56	5,56	4,63	5,41

Tabela 7.2: Continuação dos resultados comparativos da tabela 7.1 das extensões da transformada em relação ao método proposto por Hills et al. (2014).

#	TS+CS			TS+PCA		
	B. Network	R. Forest	SVM (linear)	B. network	R. Forest	SVM (linear)
1	92,58 (6)	26,59 (10)	97,44 (2)	95,91 (5)	55,50 (8)	96,42 (3)
2	70,00 (8)	53,33 (12)	63,33 (9,5)	86,67 (3,5)	80,00 (6,5)	86,67 (3,5)
4	96,95 (6)	98,54 (4,5)	99,01 (3)	89,12 (9)	91,67 (8)	98,54 (4,5)
5	96,43 (9,5)	82,14 (12)	96,43 (9,5)	96,43 (9,5)	100,00 (3,5)	100,00 (3,5)
6	61,11 (10,5)	30,72 (12)	61,11 (10,5)	89,22 (5)	97,71 (1)	90,85 (3)
8	98,49 (7,5)	50,41 (12)	98,49 (7,5)	99,77 (3)	99,88 (1,5)	97,56 (9)
10	80,68 (10,5)	44,32 (12)	80,68 (10,5)	92,05 (5)	98,86 (2)	85,23 (8)
13	92,67 (10,5)	78,00 (12)	92,67 (10,5)	100,00 (2,5)	100,00 (2,5)	96,67 (8)
14	90,28 (8)	51,51 (12)	89,12 (11)	93,78 (3)	95,72 (1)	95,63 (2)
16	69,86 (9)	73,97 (6)	71,23 (8)	80,82 (3)	82,19 (2)	72,60 (7)
17	81,97 (7)	87,11 (4)	88,68 (2)	66,05 (9)	74,87 (8)	86,84 (5)
18	85,14 (8,5)	59,82 (12)	85,14 (8,5)	94,01 (1)	93,05 (3)	84,90 (10)
21	86,02 (8,5)	47,26 (12)	86,02 (8,5)	94,51 (1,5)	94,34 (3)	84,36 (11)
23	80,16 (12)	96,67 (5)	95,67 (6)	83,00 (11)	98,33 (2)	98,00 (3)
24	98,00 (7,5)	42,00 (12)	98,00 (7,5)	100,00 (1,5)	98,00 (7,5)	99,00 (3)
25	96,75 (9,5)	88,76 (12)	96,75 (9,5)	100,00 (2,5)	100,00 (2,5)	98,86 (7)
Média	86,07	63,20	87,49	91,33	91,26	92,01
Rank Médio	8,66	10,09	7,75	4,69	3,88	5,66

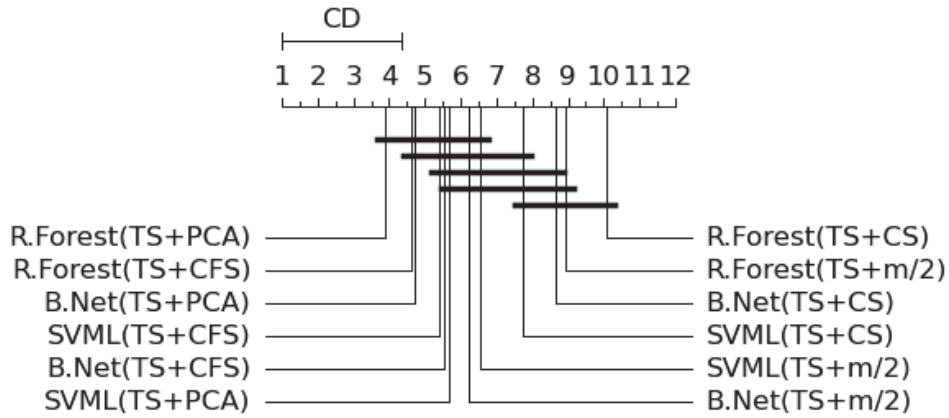


Figura 7.3: Diagrama de diferença crítica para o pós-teste de *Nemenyi*.

Por intermédio dos valores apresentados anteriormente nas tabelas 7.1 e 7.2 foi possível calcular o valor do *rank* médio das abordagens em análises nesta seção. A transformada *shapelet* ($TS+m/2$) não faz parte do *clique* das extensões $TS+PCA$ e $TS+CFS$, isto indica a existência de diferenças críticas entre essas abordagens e leva-nos a conclusão de que nestes problemas as duas extensões são mais eficientes em relação a transformada TS . Por outro lado, a extensão $TS+CS$ também não está incluída no mesmo *clique* e nem faz parte do *clique* $TS+m/2$, essas duas abordagens também apresentam diferenças críticas entre elas, neste cenário, a $TS+m/2$ é mais eficaz em relação a extensão $TS+CS$.

Por meio da aplicação do teste de *Friedman* com $\alpha = 0,05$, $K = 12$ e $N = 16$ sobre os valores de *rank* médio descritos nas tabelas 7.1 e 7.2, a hipótese nula de que todos os modelos são equivalentes em termos de acurácia foi rejeitada com valores de $F_F = 6,17$ e $p\text{-valor} = 0,0458$. Para verificar quais são as diferenças significativas em relação aos algoritmos baseados nas abordagens da transformada *shapelet* e o das extensões da transformada, o pós-teste de *Nemenyi* foi aplicado com $\alpha = 0,05$, o qual apontou valor de diferença crítica de $CD = 4,16$. A figura 7.3 descreve o diagrama de diferenças críticas.

7.3 Considerações Finais

Neste capítulo evidenciamos mais uma vez a superioridade das ETS numa comparação com a TS original. Na classificação de séries temporais usando os conceitos da primitiva *shapelet*, recomendava-se somente a utilização de classificadores ou modelos preditivos baseados em distâncias, pois os resultados de acurácia obtidos através destes modelos são difíceis de serem superados por modelos de inferências fora deste contexto. Portanto, os resultados da transformada *shapelet* e os das ETS propostas neste trabalho demonstraram que sua utilidade permite que qualquer modelo adotado apresente resultados de desempenhos significativos em termos de acurácia.

Através da tabela 7.3 e a figura 7.4 mostramos uma análise direcionada no aproveitamento das séries temporais sobre os valores de acurácias obtidos pelas extensões nos variados problemas utilizados e sumarizados anteriormente. Neste cenário, nota-se uma superioridade ou um aproveitamento nas séries temporais baseadas em *movimentos* e *simulação*.

Embora estas extensões ainda precisam ser testadas por outras medidas de maiores significâncias além da acurácia, o objetivo e a motivação que sustenta a pergunta inicial destes

estudos foram cumpridos. Entretanto, não existe uma limitação para a construção de modelos preditivos para classificação de séries temporais.

Tabela 7.3: Resultados de acurácias de acordo com os tipos de séries temporais utilizados neste estudo. Os valores estão no formato de acurácia (%) de acordo com as extensões da transformada *shapelet*.

Informações	Quantidade	Tipo de Série T	TS+CFS	TS+CS	TS+PCA
	9	Imagem	84,14	74,63	83,87
	3	Espectro	88,27	76,32	90,85
	3	Simulação	94,54	86,04	95,49
	6	Sensores	83,51	75,49	84,01
	1	Movimento	98,22	87,89	98,11
	3	ECG	92,76	86,65	93,34
Total/ Acc. Media das Média		25	90,24	81,17	90,94

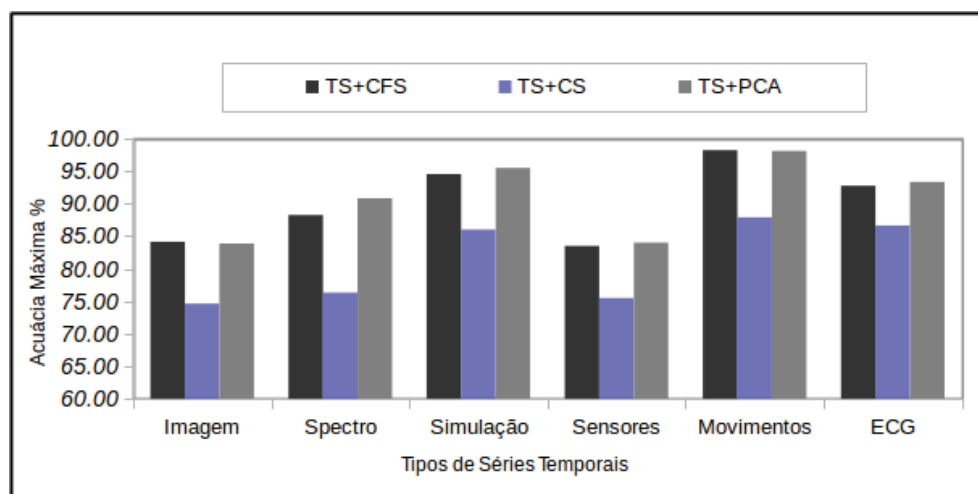


Figura 7.4: Demonstração de resultados de acurácia de acordo com os tipos de séries temporais utilizados nas extensões *TS+CFS*, *TS+CS* e *TS+PCA*.

Capítulo 8

Conclusão

Neste trabalho foi apresentada uma exploração da transformada *shapelet* contornando a desvantagem do algoritmo original proposto [Ye e Keogh \(2009\)](#) e posteriormente melhorado por [Lines et al. \(2012\)](#). Pesquisamos como poderíamos melhorar o método de seleção das melhores k *shapelet* e propusemos as extensões da transformada *shapelet* ($TS+CFS$, $TS+CS$ e $TS+PCA$), demonstramos a eficiência da descoberta das *shapelets* com características discriminatórias em m conjuntos de dados temporais e uma avaliação em termo de desempenho. mostramos, também, que nem todas as extensões são eficazes para resolver certos problemas de séries temporais originando uma dada limitação nesta abordagem.

No capítulo 6 mostramos os resultados de acurácias obtidos por cada extensão da transformada *shapelet*, bem como uma recomendação de qual algoritmo seria eficiente para ser utilizado dentro daquela mesma abordagem. Por meio destes resultados, evidenciamos a capacidade das extensões da transformada *shapelet* em prover melhores desempenhos em termos de acurácia.

No capítulo 7 realizamos uma análise individual de cada algoritmo utilizado nas extensões da transformada, onde foi constatado que a variação de acurácia do algoritmo em cada extensão pode ser reduzida através do uso da abordagem de busca de *shapelet* com heurística diferente. Ainda assim, a análise estende-se na comparação dos resultados obtidos em relação a transformada *shapelet* especificamente na metodologia proposta por [Hills et al. \(2014\)](#), e foi constatado que as extensões $TS+PCA$ e $TS+CFS$ são superiores em termo de resultados em relação à metodologia ($TS\ m/2$) considerada estado-de-arte.

No âmbito das extensões estudadas foi possível manter o processo de identificação das *shapelets* separado da fase da construção dos modelos preditivos, e também eliminamos a necessidade da estimativa dos parâmetros *mínimos e máximos* das subsequências, bem como do parâmetro k (quantidade de *shapelet*). Ainda neste contexto, demonstramos que a maneira mais fácil de utilizar a transformada *shapelet* proposta por [Lines et al. \(2012\)](#) é com a criação das extensões que possibilitam uma nova representação das séries em outro domínio com apenas uma pequena porção das melhores k *shapelets* com características discriminatórias das séries temporais.

8.1 Limitações

Conforme a nossa pesquisa, as limitações são descritas da seguinte maneira:

1. Em certos conjuntos de dados menores não é permitido o uso da extensão da transformada *shapelet*, pois, a redução brusca dos dados pode comprometer o modelo de predição, como aconteceu em vários conjuntos de dados na extensão *TS+CS*.
2. A classificação mediante a transformada ou suas extensões, não considera o domínio de tempo presente no dado. Caso haja a necessidade de compreender a variação temporal, é fundamental utilizar algoritmos ou modelos que permitam identificar estes comportamentos.
3. As ocorrências de uma determinada *shapelet* mais importante na série temporal surge como uma limitação desta pesquisa, pois, ainda não é considerada tamanha possibilidade.

8.2 Trabalhos Futuros

De acordo com as pesquisas realizadas sobre a classificação de séries temporais através da primitiva *shapelet*, são destacadas as seguintes possibilidades como futuros trabalhos:

1. A utilização de *shapelet* para a identificação de mudanças de conceitos (*concept drift*) em séries temporais.
2. Uma abordagem de seleção dinâmica de classificadores para inferências de séries temporais.
3. De acordo com a proposta de [Serra e Arcos \(2014\)](#), utilizar o método de validação-cruzada ou *holdout* com repetições para avaliar os modelos gerados considerando a união dos conjuntos de treinamento e de teste.
4. Uma análise interessante de acordo com os nossos estudos seria a de agrupamentos de séries temporais nos dados por meio da transformada *shapelets* outras medidas com maiores graus de relevâncias.

Referências Bibliográficas

- Antunes, C. M. e Oliveira, A. L. (2001). Temporal data mining: an overview. Em *KDD Workshop on Temporal Data Mining*, páginas 1–13. Citado na página 14.
- Bagnall, a. e. a. (2012). Transformation based ensembles for time series classification. in: *Proceedings of the twelfth siam international conference on data mining, anaheim, california, usa, april 26-28,*. Citado 5 vezes nas páginas 14, 37, 48, 51 e 54.
- Bakshi, B. R., Locher, G., Stephanopoulos, G. e Stephanopoulos, G. (1994). Analysis of operating data for evaluation, diagnosis and control of batch operations. *Journal of Process Control*, 4(4):179–194. Citado 2 vezes nas páginas 14 e 21.
- Batista, G. E., Wang, X. e Keogh, E. J. (2011). A complexity-invariant distance measure for time series. Em *Proceedings of the 2011 SIAM International Conference on Data Mining*, páginas 699–710. SIAM. Citado 2 vezes nas páginas 21 e 24.
- Brockwell, P. J. e Davis, R. A. (1996). *Introducing to Time Series and Forecasting*. Springer, New York, NY-USA, second edition. Citado na página 18.
- Castro, N. C. (2012). *Time series motif discovery*. Tese de doutorado, Universidade do Minho, Minho, Portugal. Citado na página 20.
- Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A. e Batista, G. (2015). The ucr time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/. Citado 3 vezes nas páginas vi, vii e 38.
- Cortes C, V. V. (1995). Support-vector networks. Citado na página 58.
- Cotofrei, P. e Stoffel, K. (2002). Rule extraction from time series databases using classification trees. Em *Proceedings of the 20th International Association of Science and Technology for Development Conference on Applied Informatics*, páginas 327–332, Innsbruck, Austria. ACTA Press. Citado na página 14.
- Cotofrei, P. e Stoffel, K. (2005). First-order logic based formalism for temporal data mining^{*}. Em *Foundations of Data Mining and knowledge Discovery*, páginas 185–210. Citado 2 vezes nas páginas 14 e 21.
- Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30. Citado na página 41.
- DING (2008). Querying and mining of time series data: experimental comparison of representations and distance measur. Citado 2 vezes nas páginas 20 e 21.

- Dunn, J. e Dunn, O. J. (1961). Multiple comparisons among means. *American Statistical Association*, páginas 52–64. Citado na página 42.
- Ehlers, R. S. (2005). Análise de séries temporais. Relatório técnico, Departamento de Estatística, Universidade Federal do Paraná, Curitiba — PR, Brasil. Citado 3 vezes nas páginas 17, 18 e 19.
- EHLERS, R. S. (2005). Análise de séries temporais. Citado na página 18.
- Esling, P. e Agon, C. (2012). Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34. Citado na página 19.
- Faceli, K., Lorena, A. C., Gama, J. e Carvalho, A. (2011). Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2:192. Citado 14 vezes nas páginas viii, 17, 21, 22, 23, 24, 25, 26, 27, 38, 39, 45, 50 e 63.
- Ferrero, C. A. (2009). Algoritmo knn para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia. Dissertação de Mestrado, Universidade de São Paulo - USP. Citado 2 vezes nas páginas 18 e 19.
- Figueiredo Filho, D. B. e Junior, J. A. S. (2010). Desvendando os mistérios do coeficiente de correlação de pearson (r). *Revista Política Hoje-ISSN: 0104-7094*, 18(1). Citado na página 23.
- Geurts, P. (2001). Pattern extraction for time series classification. Em *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, páginas 115–127, London, UK. Springer-Verlag. Citado 2 vezes nas páginas 28 e 29.
- Ghalwash, M. F. e Obradovic, Z. (2012). Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC Bioinformatics*, 13(1). Citado na página 34.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. e Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18. Citado na página 40.
- Han, J. e Kamber, M. (2006). *Data Mining: concepts and techniques*. Morgan Kaufmann, San Francisco, California, USA, 2 edition. Citado na página 21.
- Hills, J., Lines, J., Baranauskas, E., Mapp, J. e Bagnall, A. (2013). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, páginas 1–31. Citado na página 28.
- Hills, J., Lines, J., Baranauskas, E., Mapp, J. e Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881. Citado 15 vezes nas páginas 14, 15, 16, 20, 35, 36, 43, 51, 58, 64, 67, 68, 69, 70 e 73.
- Ho, T. K. (1995). Proceedings of the 3rd international conference on, document analysis and recognition, montreal, qc, 14–16 august, 1995. pp. 278–282. Citado na página 61.
- Iman, R. L. e Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics - Theory and Methods*, 9(6):571–595. Citado na página 41.
- Keogh, E. e Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371. Citado na página 20.

- Keogh, E., Lin, J. e Truppel, W. (2003). Clustering of time series subsequences is meaningless: implications for previous and future research. Em *Proceedings of the Third IEEE International Conference on Data Mining*, páginas 115–122. IEEE Computer Society. Citado 3 vezes nas páginas 20, 24 e 30.
- Keogh, E. J. e Rakthanmanon, T. (2013). Fast shapelets: A scalable algorithm for discovering time series shapelets. Em *SDM*, páginas 668–676. Citado na página 28.
- Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857 – 1874. Citado na página 21.
- Lines, J., Davis, L. M., Hills, J. e Bagnall, A. (2012). A shapelet transform for time series classification. Em *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 289–297. ACM. Citado 14 vezes nas páginas vi, vii, 16, 28, 34, 35, 36, 37, 39, 43, 48, 63, 67 e 73.
- Liu, H., Setiono, R. et al. (1996). A probabilistic approach to feature selection—a filter solution. Em *ICML*, volume 96, páginas 319–327. Citado 2 vezes nas páginas 22 e 38.
- Maletzke, A. G., Batista, G. E. d. A. P. A., Lee, H. D. e Chung, W. F. (2009). Mineração de séries temporais por meio da extração de características e da identificação de motivos. Em *Congresso da Sociedade Brasileira de Computação*. Citado na página 19.
- Morchen, F. (2003). Time series feature extraction for data mining using dwt and dft. Technical report 33, Department of Mathematics and Computer Science, Philipps-University Marburg, Germany. Citado na página 14.
- Morchen, F. (2006). *Time series knowledge mining*. Tese de doutorado, Department of Mathematics and Computer Science—Philipps-University, Marburg, Hesse, Germany. Citado 2 vezes nas páginas 17 e 19.
- Morettin, P. A. e Toloi, C. M. (2006). *Análise de Séries Temporais*. Edgard Blecher, São Paulo, Brasil, 2 edition. Citado 3 vezes nas páginas 14, 17 e 18.
- Mueen, A., Keogh, E. e Young, N. (2011). Logical-shapelets: an expressive primitive for time series classification. Em *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11*, páginas 1154–1162, New York, NY, USA. ACM. Citado na página 33.
- Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*. Princeton University. Citado na página 41.
- Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann, Califórnia, USA. Citado 2 vezes nas páginas 18 e 28.
- Rani, S. e Sikka, G. (2012). Article: Recent techniques of clustering of time series data: A survey. *International Journal of Computer Applications*, 52(15):1–9. Citado na página 21.
- Sakoe, H. e Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49. Citado na página 20.

- Serra, J. e Arcos, J. L. (2014). An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67(0):305–314. Citado na página 74.
- Silva, D. F., de Souza, V. M. A. e Batista, G. E. (2013). Time series classification using compression distance of recurrence plots. página 687–696. Citado na página 20.
- Smith, L. I. et al. (2002). A tutorial on principal components analysis. *Cornell University, USA*, 51(52):65. Citado 2 vezes nas páginas 22 e 38.
- WANG (2013). Experimental comparison of representation methods and distance measures for time series data. Citado 2 vezes nas páginas 20 e 21.
- Xing, Z., Pei, J. e Keogh, E. J. (2010). A brief survey on sequence classification. *SIGKDD Explorations*, 12(1):40–48. Citado na página 14.
- Yamada, Y., Suzuki, E., Yokoi, H. e Takabayashi, K. (2003). Decision-tree induction from time-series data based on a standard-example split test. Em *ICML*, páginas 840–847. Citado na página 29.
- Ye, L. e Keogh, E. (2009). Time series shapelets: a new primitive for data mining. Em *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 947–956. ACM. Citado 7 vezes nas páginas 14, 28, 29, 31, 34, 36 e 73.
- Ye, L. e Keogh, E. J. (2011). Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Discov.*, 22(1-2):149–182. Citado 3 vezes nas páginas 31, 32 e 51.
- Zalewski, W. (2015). Modelagem simbólica de padrões morfológicos para classificação de séries temporais. Citado 11 vezes nas páginas 17, 19, 20, 21, 22, 26, 31, 32, 35, 48 e 54.
- Zalewski, W., Silva, F., Maletzke, A. e Ferrero, C. (2016). Exploring shapelet transformation for time series classification in decision trees. *Knowledge-Based Systems*, 112:80 – 91. Citado 13 vezes nas páginas vi, vii, 14, 15, 17, 22, 23, 30, 36, 37, 43, 51 e 63.